6.- ANÁLISIS DE EFECTOS DE LA PRECISIÓN FINITA EN FILTROS DIGITALES.

6.1.- INTRODUCCIÓN.

Los filtros considerados hasta ahora eran "filtros discretos" pero no filtros digitales, en el sentido que las señales de entrada, estaban cuantizadas en tiempo pero no en amplitud y los coeficientes no estaban cuantizados. Estos sistemas, al ser implementados en *hardware* o *software*, deben cuantizar dichos valores a los permitidos por la representación numérica utilizada. Este proceso transforma al filtro en un sistema NO LINEAL. En general estos efectos son difíciles de analizar, si bien, si su contribución es pequeña comparada con las señales, pueden ser considerados como perturbaciones aditivas a un sistema lineal, lo que permite la utilización de técnicas estadísticas para evaluar sus efectos. Esto permite obtener estimadores que posteriormente se compararán con los resultados experimentales. Las fuentes de error son las siguientes:

- Conversión AD
- Cuantización de los coeficientes del filtro.
- Cuantización de las operaciones aritméticas.
- Presencia de Ciclos Límite

Las representaciones numéricas de los datos pueden ser en coma fija o en coma flotante, en cualquier caso se realiza una representación con un número finito de bits, lo cual se traduce en que se produzcan efectos no deseados. Así, por ejemplo, el producto de dos números representados con *b* bits produce un resultado de longitud *2b* que posteriormente deberá ser cuantizado para su almacenamiento en un registro de *b* bits. Además, en aritmética de coma fija, al sumar dos magnitudes de b bits se puede obtener un resultado que exceda el valor máximo representado, produciéndose un error. Es decir, el análisis de los efectos de la cuantización en un filtro digital depende de varios parámetros:

- Formato de los datos (coma fija, coma flotante).
- Tipo de representación numérica utilizada (signo magnitud, complemento 2, etc.)
- Tipo de cuantización (redondeo, truncamiento)
- Estructura utilizada para la implementación del filtro.

Analicemos cada uno de estos parámetros por separado.

6.2.- Representación numérica en coma fija (Punto fijo).

En general un número en punto fijo puede representarse como:

$$x = (b_{-A},...,b_{-1},b_0,b_1,...,b_B)_r = \sum_{i=-A}^{B} b_i r^{-i} \qquad 0 \le b_i \le (r-1)$$

A: es el número de dígitos para la parte entera

B: es el número de dígitos para la parte fraccionaria.

r: es la base

r=2→binario, b_i= 0,1

r=10 → decimal, b_i= 0,1,…,9

r=16 → hexadecimal b_i= 0,1,…,9,A,B,C,D,E,F

Nos vamos a centrar en la representación binaria *r=2*, ya que es la utilizada por los dispositivos hardware. En este caso los dígitos se denominan *bits* (**b**inary dig**it**).

b_{-A} : *bit más significativo (MSB)*

b_B : *bit menos significativo (LSB)*

Si

$A = n-1 \quad B = 0 \rightarrow$ Formato entero sin signo. Intervalo: $0,...,2^n - 1$

$A = 0 \quad B = n-1 \rightarrow$ Formato fraccional. Intervalo: $0,...,1 - 2^{-n}$

En general, un número con parte entera y parte fraccional se representará asignado un número de bits para cada parte. En lo sucesivo consideraremos el formato fraccional binario.

Los números fraccionales binarios positivos tienen un formato definido por:

$$x = 0.b_1,...b_B = \sum_{i=1}^{B} b_i 2^{-i} \qquad x > 0$$

El MSB=0 indica signo positivo.

Para los números negativos hay tres formatos de representación:

Signo Magnitud: el MSB se pone a 1 para indicar signo negativo.

$x_{SM} = 1.b_1,...b_B \qquad x \le 0$

Complemento a 1.

$x_{C1} = 1.\overline{b_1},...\overline{b_B} \qquad x \le 0$

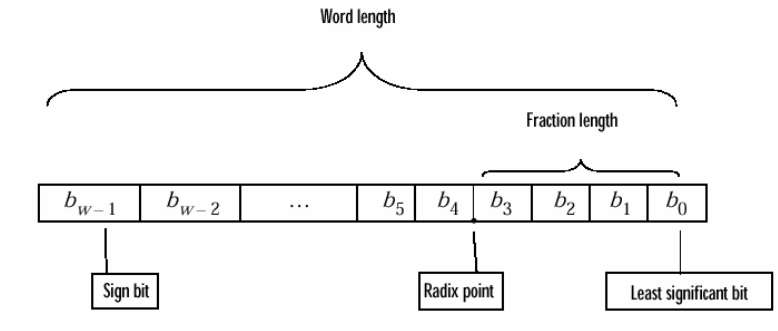
$\overline{b_1}$ es el complemento a 1 del bit (cambiar unos por ceros y viceversa) o alternativamente se puede calcular como $x_{C1} = 2 - 2^{-B} - |x|$

Complemento a 2. Se obtiene a partir del complemento a 1 sumándole un LSB

$x_2 = 1.\overline{b_1},...\overline{b_B} + LSB \qquad x \le 0$

Una definición alternativa es: $x_{C2} = 2 - |x|$.

Observamos que dada una secuencia de bits, el número que debemos interpretar depende de la representación utilizada. De las indicadas la más usual es el complemento a 2.



- 10110 represents the integer $-2^4+2^2+2 = -10$.
- 10.110 represents $-2+2^{-1}+2^{-2} = -1.25$.
- 1.0110 represents $-2^{-0}+2^{-2}+2^{-3} = -0.625$.

Veamos a continuación algunas definiciones:

Rango dinámico: es la diferencia entre el número más grande y el más pequeño en una determinada representación. $R = x_{\max} - x_{\min}$

Precisión o Resolución: Mínima distancia entre dos número consecutivos representados dentro del rango dinámico. $\Delta = \frac{x_{\max} - x_{\min}}{2^B - 1}$, B: número total de bits

Ejemplo:

Si tenemos B+1 bits (B bits significativos más un bit de signo) y empleamos representaciones fraccionarias en C2 el intervalo de números representado será: $-1 \le x \le 1 - 2^{-B}$

Luego $R = 2 - 2^{-B}$ $\quad \Delta = 2^{-B}$

La resolución en la representación en coma fija es constante.

Desbordamiento: es el efecto producido cuando tenemos un número que está fuera del rango dinámico para una representación específica. Puede ser por exceso (*overflow*) o por defecto (*underflow*).

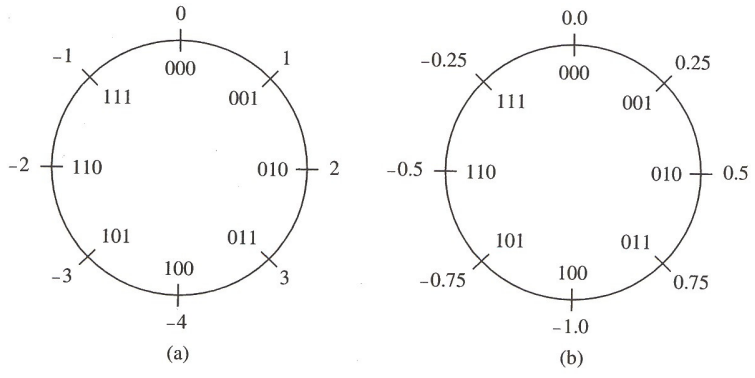
La denominación de punto fijo se debe a que el punto decimal está siempre en la misma posición, una representación alternativa es la representación en punto flotante que veremos a continuación.

En la siguiente tabla se muestra la representación binaria fraccional con 3 bits significativos más uno de signo, en los diferentes formatos de representación vistos anteriormente.

Decimal equivalent	Sign-magnitude	Ones'-complement	Two's-complement	Offset binary
7/8	$0_\Delta 111$	$0_\Delta 111$	$0_\Delta 111$	$1_\Delta 111$
6/8	$0_\Delta 110$	$0_\Delta 110$	$0_\Delta 110$	$1_\Delta 110$
5/8	$0_\Delta 101$	$0_\Delta 101$	$0_\Delta 101$	$1_\Delta 101$
4/8	$0_\Delta 100$	$0_\Delta 100$	$0_\Delta 100$	$1_\Delta 100$
3/8	$0_\Delta 011$	$0_\Delta 011$	$0_\Delta 011$	$1_\Delta 011$
2/8	$0_\Delta 010$	$0_\Delta 010$	$0_\Delta 010$	$1_\Delta 010$
1/8	$0_\Delta 001$	$0_\Delta 001$	$0_\Delta 001$	$1_\Delta 001$
0	$0_\Delta 000$	$0_\Delta 000$	$0_\Delta 000$	$1_\Delta 000$
−0	$1_\Delta 000$	$1_\Delta 111$	N/A	N/A
−1/8	$1_\Delta 001$	$1_\Delta 110$	$1_\Delta 111$	$0_\Delta 111$
−2/8	$1_\Delta 010$	$1_\Delta 101$	$1_\Delta 110$	$0_\Delta 110$
−3/8	$1_\Delta 011$	$1_\Delta 100$	$1_\Delta 101$	$0_\Delta 101$
−4/8	$1_\Delta 100$	$1_\Delta 011$	$1_\Delta 100$	$0_\Delta 100$
−5/8	$1_\Delta 101$	$1_\Delta 010$	$1_\Delta 011$	$0_\Delta 011$
−6/8	$1_\Delta 110$	$1_\Delta 001$	$1_\Delta 010$	$0_\Delta 010$
−7/8	$1_\Delta 111$	$1_\Delta 000$	$1_\Delta 001$	$0_\Delta 001$
−8/8	N/A	N/A	$1_\Delta 000$	$0_\Delta 000$

Extraído de: Tratamiento Digital de Señales. J.G. Proakis

La figura siguiente muestra los valores posibles en una representación de 3 bits en complemento a 2 (a) se corresponde con una representación entera y (b) con una fraccional. Las secuencias de bits son idénticas, únicamente cambia la interpretación.



Extraído de: Tratamiento Digital de Señales. J.G. Proakis

6.3.- Representación numérica en coma flotante.

La representación en coma flotante es de la forma:

$$x = M \cdot 2^E \qquad \begin{matrix} M: \text{mantisa} \quad 0.5 \le M < 1 \\ E: \text{exponente} \end{matrix}$$

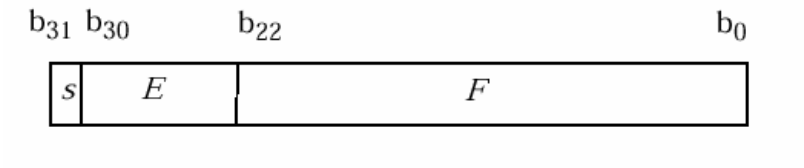
$$\pm d.dddd \times 10^{p} = \pm ddddd.0 \times 10^{p-4} = \pm 0.ddddd \times 10^{p+1}$$

$$\pm b.bbbb \times 2^{q} = \pm bbbbb.0 \times 2^{q-4} = \pm 0.bbbbb \times 2^{q+1}$$

El exponente puede cambiar para mantener la mantisa en el intervalo permitido (COMA FLOTANTE)

Para la mantisa y el exponente se puede utilizar cualquiera de las representaciones vistas (signo magnitud, complemento-1, complemento-2).

Existe un estándar para la representación numérica en coma flotante (IEEE 754). Para precisión simple[1] (32 bits), los bits están distribuidos de la siguiente manera:



s: signo (1 bit)

F: mantisa (23 bits)

E: exponente (8 bits). El exponente también incluye un bit de signo.

La siguiente tabla muestra los valores máximo y mínimo en valor absoluto que podemos representar con aritmética de coma flotante

	Signo	M (23 bits)	Signo (E)	E(7bits	
Mínimo	0	1000…..0	1	1111111	$0.5 \cdot 2^{-127} \approx 0.3 \cdot 10^{-38}$
Máximo	1	1111…...1	0	1111111	$(1-2^{-23}) \cdot 2^{127} \approx 1.7 \cdot 10^{38}$

Si consideramos aritmética de punto fijo de 32 bits, con un bit de signo tenemos, en valores absolutos

[1] Existe un formato con precisión doble de 64 bits con M=52, E=11 y un bit de signo.

	Signo	31 bits			
Mínimo	-	00000….1			$2^{-31} \approx 4.6 \cdot 10^{-10}$
Máximo	-	11111….1			$2^{32} - 1 \approx 4.6 \cdot 10^{9}$

La representación en coma flotante tiene un rango dinámico mayor, que consigue mediante una resolución variable. La resolución es fina para números pequeños y gruesa para números grandes. Para la representación en coma fija la precisión es constante. Si tenemos en cuenta, solo la parte significativa; mantisa, para un mismo número de bits, la precisión en coma fija es mayor que en coma flotante.

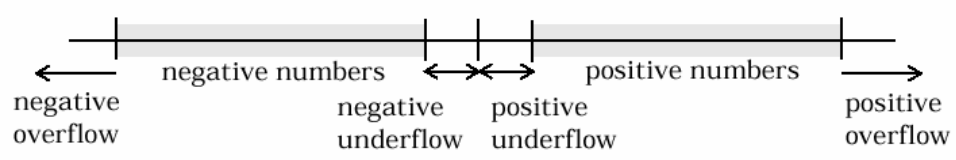
Existen algunas combinaciones determinadas de E y M que sirven para almacenar resultados "especiales" de las operaciones:

Si E=255 y M≠0	X no es un número
Si E=255 y M=0	$X=(-1)^s\infty$
Si 0<E<255	$X==(-1)^s\ 2^{E-127}(1.M)$
Si E=0 y M≠0	$X==(-1)^s\ 2^{-126}\ (0.M)$
Si E=0 y M=0	$X==(-1)^s\ 0$

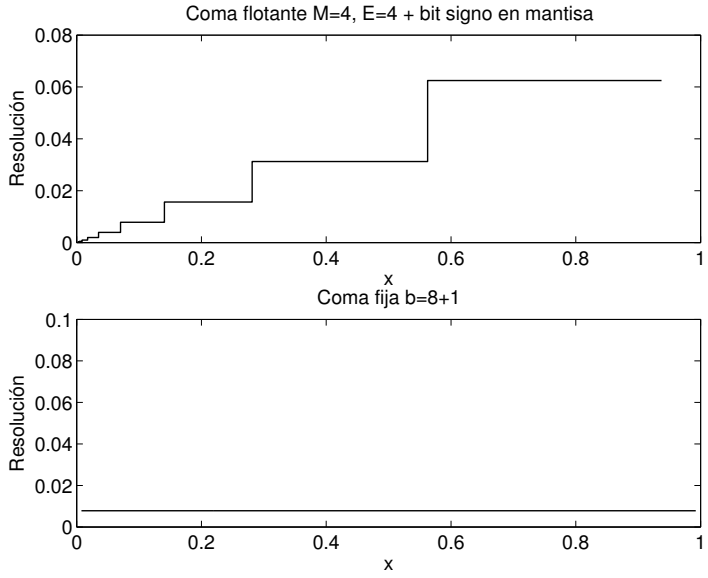
0.M es una representación fraccionaria y 1.M es una representación mixta (tiene parte entera y parte fraccionaria) con 1 bit entero.

En la siguiente tabla mostramos el intervalo de representación y precisión en la representación en coma flotante con precisión simple y doble

Floating-Point Data Type	Normalized Minimum	Maximum	Exponent Bias	Precision
Single	$2^{-126}\approx 10^{-38}$	$(2-2^{-23})2^{127}\approx 3(10^{38})$	127	$2^{-23}\approx 10^{-7}$
Double	$2^{-1022}\approx 2(10^{-308})$	$(2-2^{-52})2^{1023}\approx 1.7(10^{308})$	1023	$2^{-52}\approx 10^{-16}$
Custom	2^{1-bias}	$(2-2^{-f})2^{bias}$	$2^{e-1}-1$	2^{-f}

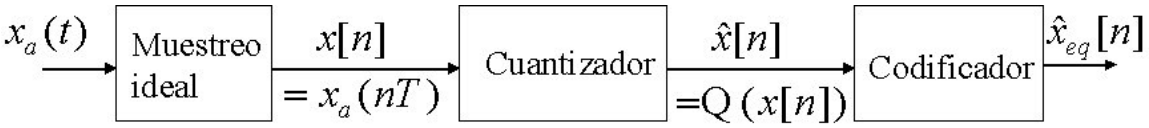


A continuación mostramos gráficamente la resolución en una representación en coma fija fraccional de 8 bits significativos más uno de signo y en una representación en coma flotante con 4 bits para la mantisa y 4 para el exponente más un bit de signo adicional para la mantisa. Se pone de manifiesto como la precisión en coma fija es constante mientras que en coma flotante ésta varía con cada modificación del valor del exponente, de manera que ésta se va duplicando; es decir, la precisión para números pequeños es muy buena y empeora para números grandes.



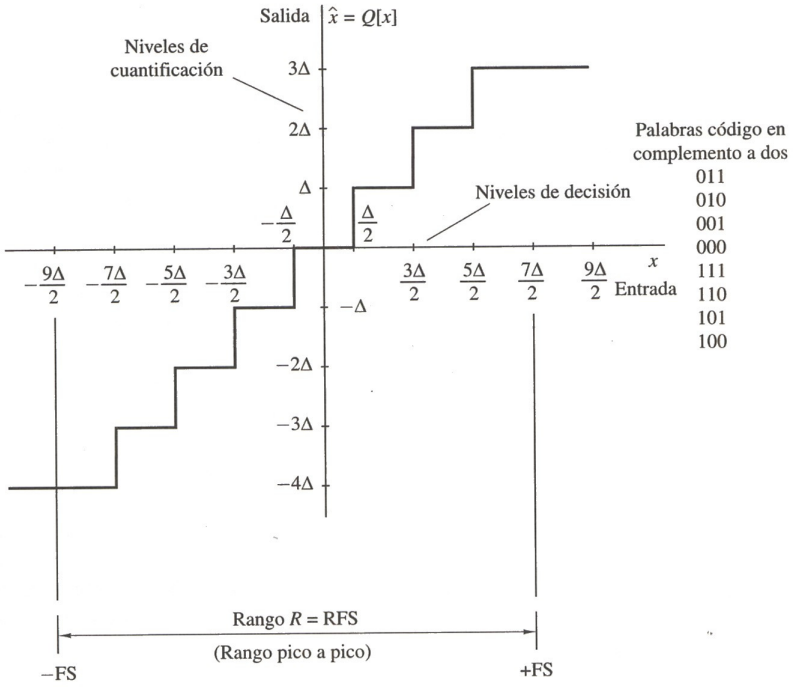
6.4.- *Cuantización en la conversión AD.*

Sabemos que la conversión analógica digital es un proceso que permite transformar una señal continua en tiempo y amplitud en una señal que es discreta en tiempo y amplitud. La figura siguiente muestra todo el proceso.



La muestra digital final es la representación binaria de la señal cuantizada producida por un muestreador ideal. Si la salida se representa con b bits, incluyendo el bit de signo, el número de niveles digitales posibles es 2^b.

El proceso de cuantización puede realizarse por **redondeo** (asignación al nivel más próximo) o por **truncamiento** (asignación al nivel inmediatamente inferior). La función de transferencia, para un conversor de 3 bits por redondeo se muestra en la siguiente gráfica.



Extraído de: Tratamiento Digital de Señales. J.G. Proakis

Si el conversor funciona por truncamiento la gráfica es idéntica pero con un desplazamiento $\frac{\Delta}{2}$ a la derecha, con $\Delta = 2^{-b}$.

El valor equivalente para la muestra cuantizada, en una representación binaria fraccional en complemento a 2, para la muestra $\hat{x}[n]$, es $-1 \le \hat{x}_q[n] < 1$, que está relacionada con la

muestra cuantizada como $\hat{x}_{eq}[n] = \frac{2\hat{x}[n]}{R_{FS}}$, siendo R_{FS} el intervalo de entrada del AD. Es decir, hemos realizado un escalado por $R_{FS}/2$ para obtener una entrada en el intervalo ±1.

La diferencia entre la señal original y la cuantizada se denomina **error de cuantización**. Los errores cometidos cuando se utiliza redondeo o truncamiento con *b* bits son:

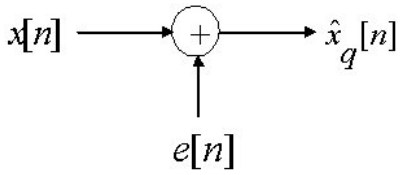
$$e(n) = \hat{x}_q(n) - x(n) \qquad \text{Error de cuantización}$$

$$\text{Redondeo} \quad -\frac{\Delta}{2} < e[n] \le \frac{\Delta}{2} \qquad \Delta = 2^{-b}$$

$$\text{Truncamiento} \quad -\Delta < e[n] \le 0 \qquad \Delta = 2^{-b}$$

Si la señal de entrada excede el rango del conversor, se produce un error en la conversión que aumenta linealmente con la señal de entrada, es lo que se denomina **ruido de saturación o sobrecarga.**

El proceso de conversión AD es un proceso no lineal, y no invertible en el que siempre hay una pérdida de información. Este proceso se modeliza como:



Extraído de: Digital Signal Processing. A computer-based approach. S. K, Mitra.

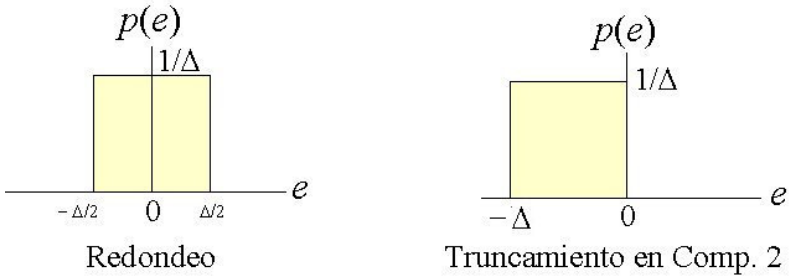
Siendo x(n) la señal sin cuantizar y e(n) una variable aleatoria con las siguientes características:

1. La secuencia de error es una versión muestreada de un proceso de ruido blanco estacionario en sentido amplio (wide-sense stationary) en el que cada muestra esta uniformemente distribuida en el intervalo determinado por el error de cuantización.
2. La secuencia de error no está correlacionada con la secuencia de entrada correspondiente.
3. La secuencia de entrada es una secuencia procedente de un proceso aleatorio estacionario.

Estas condiciones se verifican si las señales de entrada tienen una amplitud grande, comparada con el escalón de cuantización, ya que en este caso, la variación de la amplitud se puede considerar aleatoria.

Cuando se utiliza una representación en complemento a 1 o signo-magnitud con truncamiento, la señal de error sí está correlacionado con la señal de entrada ya que el error obtenido es siempre de signo contrario al signo de la señal de entrada, por esta razón una representación en complemento a 2 con cuantización por truncamiento o una cuantización por redondeo es preferible.

Las funciones densidad de probabilidad del error por redondeo y truncamiento (en una representación en complemento a dos) son las siguientes



Extraído de: Digital Signal Processing. A computer-based approach. S. K, Mitra.

A partir de la funciones densidad de probabilidad podemos calcular valores medios y varianzas:

$$\bar{e} = E\{e\} = \int e \cdot p(e) \cdot de$$

$$\sigma_e^2 = E\{e^2\} - (E\{e\})^2 = \int e^2 \cdot p(e) \cdot de - \left(\int e \cdot p(e) \cdot de\right)^2$$

Obtenemos:

$$\text{Redondeo}: \begin{matrix} \bar{e}_r = 0 \\ \sigma_r^2 = \dfrac{\Delta^2}{12} \end{matrix} \qquad \text{Truncamiento}: \begin{matrix} \bar{e}_t = -\dfrac{\Delta}{2} \\ \sigma_t^2 = \dfrac{\Delta^2}{12} \end{matrix}$$

El efecto de la conversión AD en la relación señal ruido se mide con la relación señal ruido de cuantización definido como:

$$SNR_{AD} = 10 \cdot \log \frac{Energia\ señal}{Energia\ ruido}$$

si $Energia\ ruido = \sigma^2$ para un conversor de b bits, ($\Delta = 2^{-b}$) obtenemos:

$$SNR_{AD} = \left(10 \cdot \log\left(\sigma_x^2\right) + 10.8 + 6.02b\right) dB$$

Cada bit adicional proporciona un aumento teórico de 6 bits en la SNR. El resultado obtenido depende de las características de la señal de entrada. Para un conversor bipolar con rango de entrada *2A* y *b* bits, con una entrada sinusoidal de amplitud *A*, la expresión obtenida es:

$$SNR_{AD} = 10 \cdot \log \frac{Energia\ señal}{Energia\ ruido} = 10 \log \frac{\frac{A^2}{2}}{\left(\left(\frac{2A}{2^b}\right)^2 \cdot \frac{1}{12}\right)} = \left(6.03b + 1.76\right) dB$$

El ruido generado en la conversión AD, con varianza σ_{AD}^2, al atravesar el filtro genera un ruido a la salida cuya varianza viene dada por:

$$\boxed{\sigma_{oAD}^2 = \sigma_{AD}^2 \sum_{k=0}^{\infty} h^2(k)}$$

6.5.- Cuantización de las operaciones aritméticas:errores de redondeo y truncamiento.

Durante la realización de operaciones aritméticas, por ejemplo al multiplicar 2 números representados con b bits, si el resultado de 2b bits queremos almacenarlo en el mismo formato deberemos realizar una reducción de *2b* → *b* bits. Esta reducción implica una cuantización, que puede realizarse por redondeo o truncamiento.

Cuantización de operaciones: $x = \overbrace{0.10110...01}^{b_0}$ $\quad Q(x) = \overbrace{0.10110...11}^{b}$ $\quad b < b_0$

Dado que los número positivos tienen la misma representación en los tres tipos de numeración analizados, únicamente estudiaremos por separado los números negativos. En la siguiente tabla se muestra los errores de redondeo y truncamiento en cada uno de los casos. Si $b_0 >>> b$ el término 2^{-b_0} puede despreciarse.

$$\text{COMA FIJA}\ \ b_0 \to b,\quad E = Q(x) - x \begin{cases} \text{Truncamiento} \begin{cases} \text{Positivos y Negat. en Comp. 2} & \to -\left(2^{-b} - 2^{-b_0}\right) \le E_t \le 0 \\ \text{Negat. en Comp. 1 y Negat. Signo - Mag} & \to 0 \le E_t \le \left(2^{-b} - 2^{-b_0}\right) \end{cases} \\ \text{Redondeo} \quad \text{(Todas las representaciones)} & \to -\dfrac{\left(2^{-b} - 2^{-b_0}\right)}{2} \le E_r \le \dfrac{\left(2^{-b} - 2^{-b_0}\right)}{2} \end{cases}$$

Ejemplos[2]:

$$SM: x_{decimal} = +0.6875 \quad x_{bin} = 0.10110 \xrightarrow{trunc.a2bits} Q(x)_{bin} = 0.10 \quad Q(x)_{decimal} = 0.50 \quad E_t = -0.1875$$
$$SM: x_{decimal} = -0.6875 \quad x_{bin} = 1.10110 \xrightarrow{trunc.a2bits} Q(x)_{bin} = 1.10 \quad Q(x)_{decimal} = 0.50 \quad E_t = +0.1875$$
$$C2: x_{decimal} = +0.6875 \quad x_{bin} = 0.10110 \xrightarrow{trunc.a2bits} Q(x)_{bin} = 0.10 \quad Q(x)_{decimal} = 0.50 \quad E_t = -0.1875$$
$$C2: x_{decimal} = -0.6875 \quad x_{bin} = 1.01010 \xrightarrow{trunc.a2bits} Q(x)_{bin} = 1.01 \quad Q(x)_{decimal} = -0.75 \quad E_t = -0.0625$$
$$SM: x_{decimal} = +0.6875 \quad x_{bin} = 0.10110 \xrightarrow{red.a2bits} Q(x)_{bin} = 0.11 \quad Q(x)_{decimal} = 0.75 \quad E_r = +0.0625$$
$$SM: x_{decimal} = -0.6875 \quad x_{bin} = 1.10110 \xrightarrow{red.a2bits} Q(x)_{bin} = 1.11 \quad Q(x)_{decimal} = -0.75 \quad E_r = -0.0625$$
$$C2: x_{decimal} = +0.6875 \quad x_{bin} = 0.10110 \xrightarrow{red.a2bits} Q(x)_{bin} = 0.11 \quad Q(x)_{decimal} = 0.75 \quad E_r = +0.0625$$
$$C2: x_{decimal} = -0.6875 \quad x_{bin} = 1.01010 \xrightarrow{red.a2bits} Q(x)_{bin} = 1.01 \quad Q(x)_{decimal} = -0.75 \quad E_r = -0.0625$$

Cuando se utiliza aritmética de coma flotante se cuantiza únicamente la mantisa M. Dado que la resolución en coma flotante no es uniforme sino de depende del número que se cuantifica, se define un error relativo

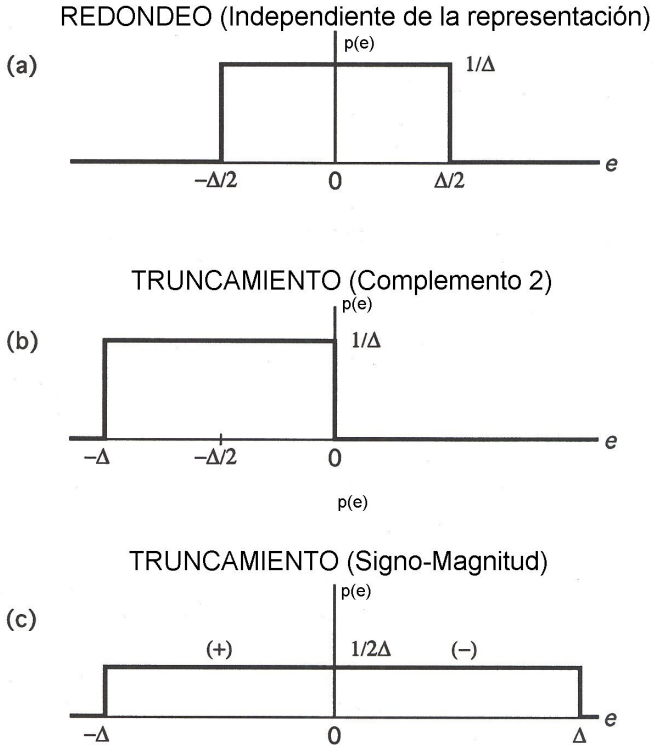
$$e = \frac{Q(x) - x}{x} = \frac{Q(M) - M}{M}$$

$$\text{COMA FLOTANTE.}\ \ b_0 \to b,\ b_0 = \infty,\ x = M \cdot 2^E,\ M\ (b \text{ bits}) \begin{cases} \text{Truncamiento} \begin{cases} \text{Comp. 2} & \to & -2 \cdot 2^{-b} < e_t \le 0, \quad x > 0 \\ & & 0 \le e_t < 2 \cdot 2^{-b}, \quad x < 0 \\ \text{Comp. 1 y Signo - Mag} & \to & -2 \cdot 2^{-b} < e_t \le 0 \end{cases} \\ \text{Redondeo} \quad \text{(Todas las representaciones)} & & \to -2^{-b} < e_r \le 2^{-b} \end{cases}$$

En lo sucesivo vamos a considerar únicamente sistemas de coma fija.

[2] La interpretación de un número negativo, con representación fraccional, en complemento a 2 en decimal es $x_{c2} = -(2 - |x|)$. Ej $x_{c2} = 1.01 \qquad x_{dec} = -(2 - (2^0 + 2^{-2})) = -0.75$

En la siguiente figura se muestra las gráficas de las funciones densidad de probabilidad de los errores por redondeo y truncamiento, utilizando representaciones en complemento a 2 y signo magnitud.



Para los casos *(a)* y *(b)* ya hemos calculado los valores medios y las varianzas. Para el caso *(c)* obtenemos:

$$\text{Truncamiento(signo - magnitud)}: \begin{matrix} \bar{e}_t = 0 \\ \sigma_t^2 = \dfrac{\Delta^2}{3} \end{matrix}$$

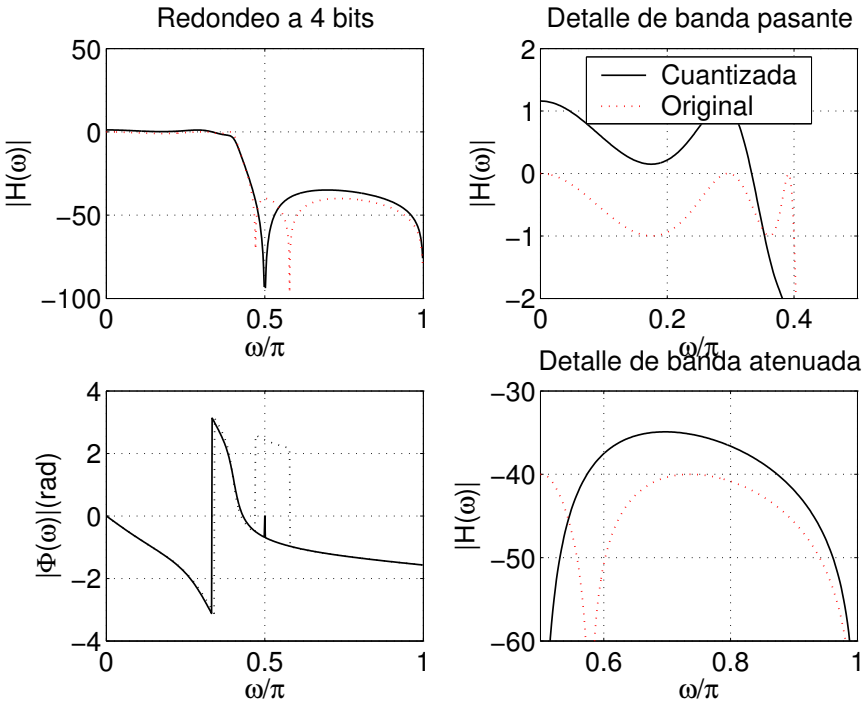
Si comparamos con los valores obtenidos para cuantización por redondeo observamos que la varianza (error) es 4 veces superior. Además el error de truncamiento está correlacionado con la señal de entrada ya que si $x > 0 \quad e_t < 0$ y $x < 0 \quad e_t > 0$; es decir, no se verifica uno de los requisitos que habíamos impuesto originalmente. Por otra parte, una ventaja de esta representación es que no aparece un efecto que veremos más adelante como son los **ciclos límite**.

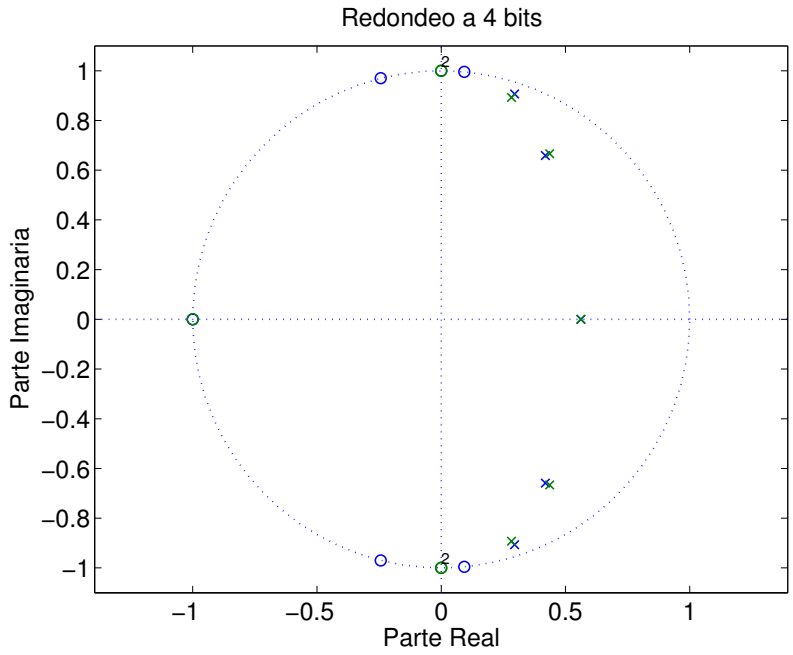
6.6.- Cuantización de los coeficientes de un filtro digital.

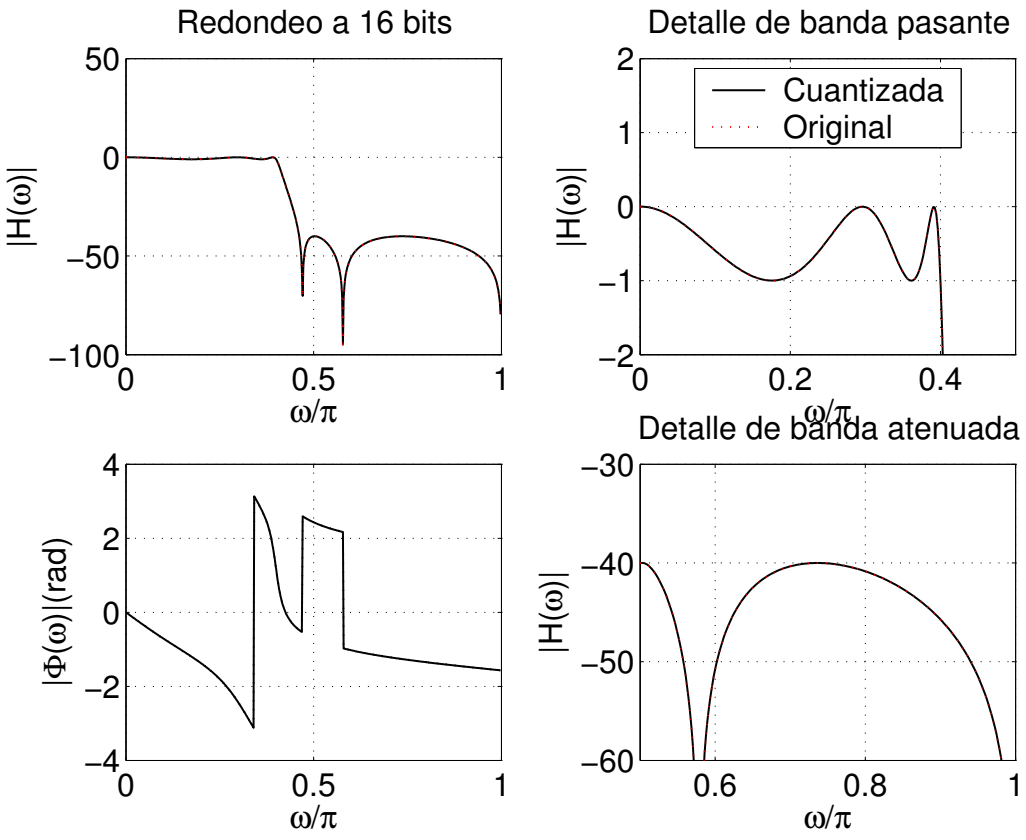
La función de transferencia $\overline{H(z)}$ de un filtro digital obtenidos al cuantizar los coeficientes, es diferente de la función de transferencia del filtro deseado $H(z)$. El principal efecto es la modificación de la localización de los ceros y los polos del filtro original. Pudiendo incluso hacer que un filtro estable dé lugar a un filtro inestable. Además se modifica la respuesta en frecuencia de filtro con lo que el nuevo sistema puede no cumplir las especificaciones de diseño del filtro original. Veámoslo en las siguientes gráficas.

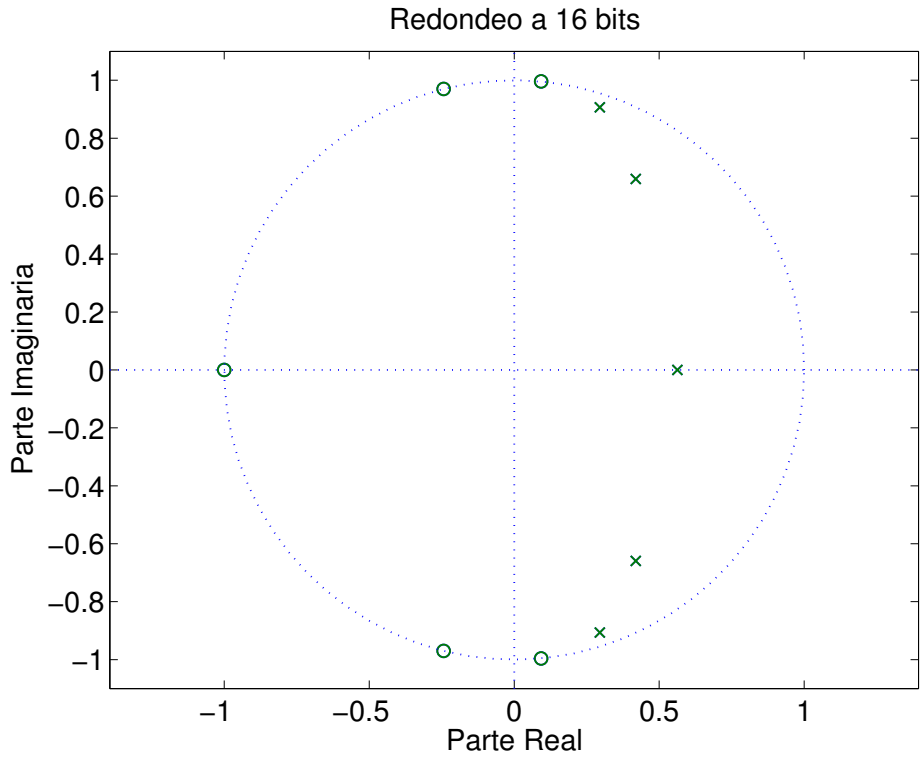
Se ha diseñado un filtro elíptico, con coef. cuantizados por redondeo, con las instrucciones:

```
Rp=1;Rs=40;Wp=0.4;Ws=0.5;
[N, Wn] = ellipord(Wp, Ws, Rp, Rs);
[b,a]=ellip(N,Rp,Rs,Wn);
```


Redondeo a 4 bits
|H(ω)|
ω/π
Detalle de banda pasante
Cuantizada
Original
|H(ω)|
ω/π
|Φ(ω)|(rad)
ω/π
Detalle de banda atenuada
|H(ω)|
ω/π

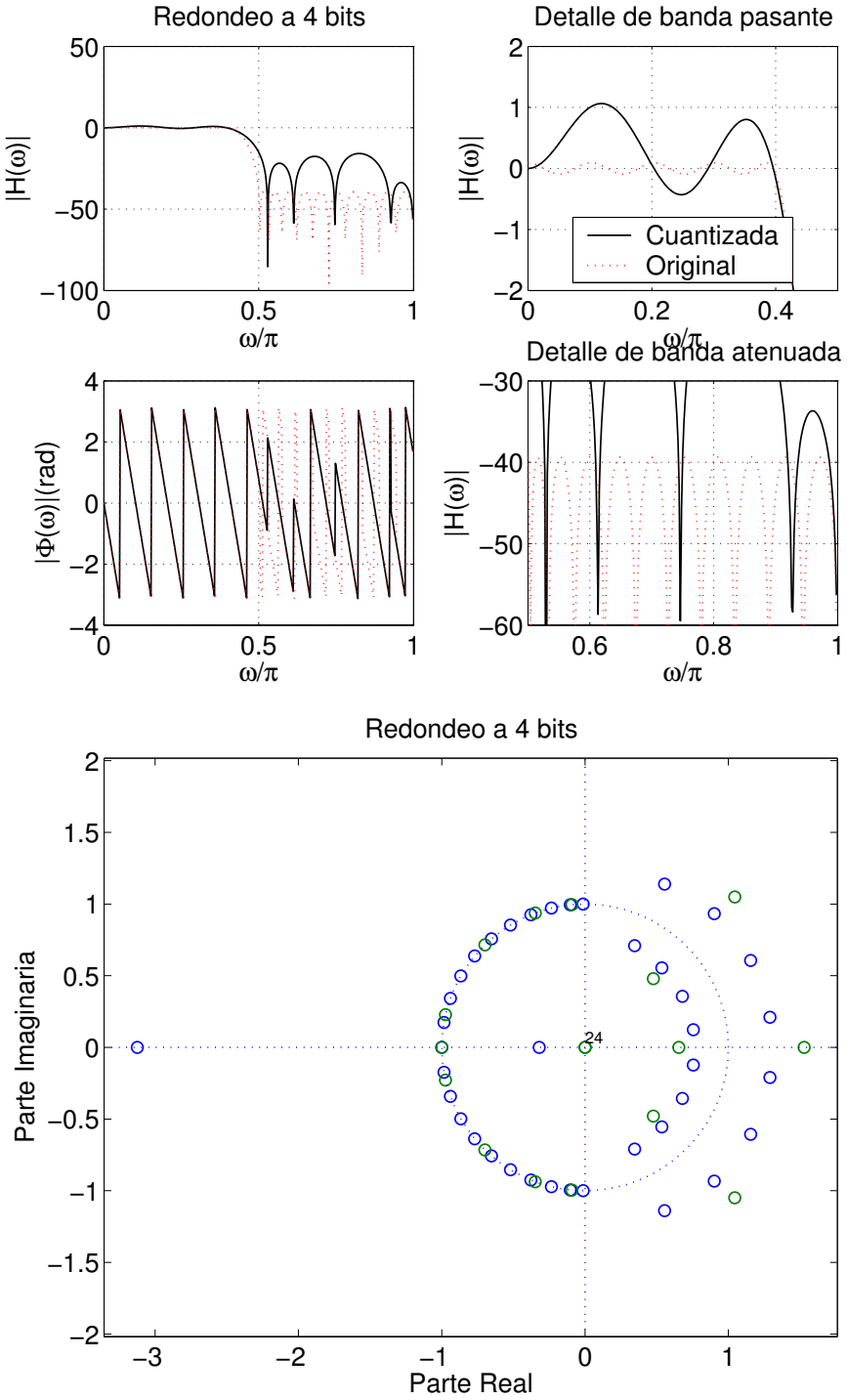

Redondeo a 4 bits
Parte Imaginaria
Parte Real

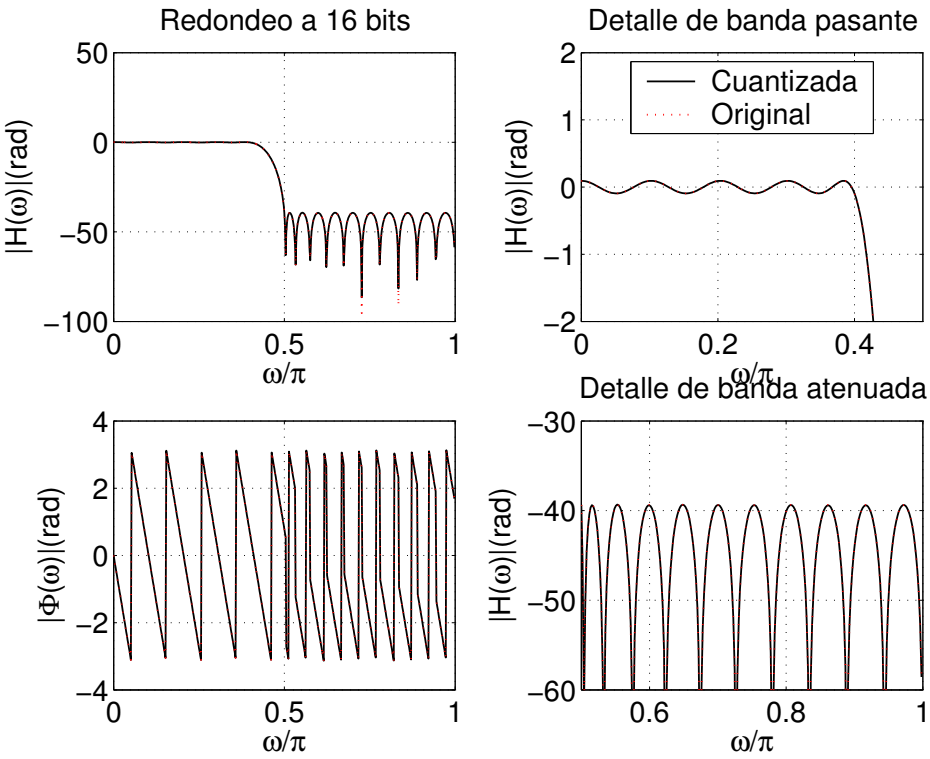

Redondeo a 16 bits
|H(ω)|
ω/π
Detalle de banda pasante
Cuantizada
Original
|H(ω)|
ω/π
|Φ(ω)|(rad)
ω/π
Detalle de banda atenuada
|H(ω)|
ω/π

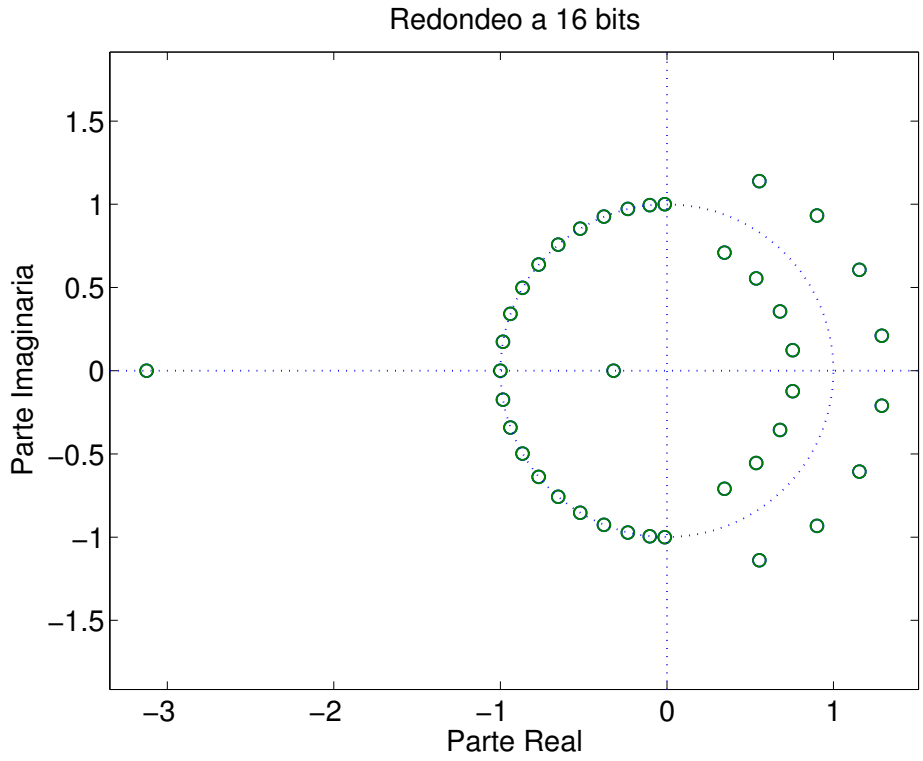

Redondeo a 16 bits
Parte Imaginaria
Parte Real

Ahora repetimos el proceso para un filtro FIR de similares características, cuantizando los coeficientes por redondeo. Se ha utilizado el siguiente código:

```
[n,fo,mo,w]=remezord([0.4 0.5],[1 0],[0.01 0.01]);b= remez(n,fo,mo,w);
```




Redondeo a 16 bits
|H(ω)|(rad)
ω/π
Detalle de banda pasante
Cuantizada
Original
|H(ω)|(rad)
ω/π
|Φ(ω)|(rad)
ω/π
Detalle de banda atenuada
|H(ω)|(rad)
ω/π


Redondeo a 16 bits
Parte Imaginaria
Parte Real

6.6.1.- Sensibilidad frente a la cuantización de los polos.

En este apartado vamos a analizar las modificaciones que experimentan los polos como consecuencia de cuantizar los coeficientes del filtro. Dado un filtro IIR

$$H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}}$$

Consideremos que al cuantizar los coeficientes tenemos:

$$\begin{aligned}\overline{b_k} &= b_k + \Delta b_k \\ \overline{a_k} &= a_k + \Delta a_k\end{aligned}$$ siendo Δb_k y Δa_k las perturbaciones (errores) de cuantización.

La nueva función de transferencia del filtro será:

$$\overline{H(z)}=\frac{\sum_{k=0}^{M} \overline{b_k} z^{-k}}{1+\sum_{k=1}^{N} \overline{a_k} z^{-k}}$$

Analicemos los polos. Sea $D(z)=1+\sum_{k=1}^{N} a_k z^{-k}=\prod_{k=1}^{N}\left(1-p_k z^{-1}\right)$ y tras la cuantización

$$\overline{D(z)}=\prod_{k=1}^{N}\left(1-\overline{p_k} z^{-1}\right) \qquad \overline{p_k}=p_k+\Delta p_k$$

Vamos a relacionar Δp_k con Δa_k, para ello expresamos que la variación total experimentada por un polo es igual a la suma de las variaciones de dicho polo respecto de cada uno de los coeficientes del filtro y lo expresamos como:

$$\Delta p_i=\sum_{k=1}^{N}\frac{\partial p_i}{\partial a_k}\Delta a_k$$

Expresión que podemos calcular a partir de *D(z)* aplicando la regla de la cadena,

$$\left.\frac{\partial D(z)}{\partial a_k}\right|_{z=p_i}=\left.\frac{\partial D(z)}{\partial z}\right|_{z=p_i}\left.\frac{\partial z}{\partial a_k}\right|_{z=p_i} \Rightarrow \frac{\partial p_i}{\partial a_k}=\frac{\left.\frac{\partial D(z)}{\partial a_k}\right|_{z=p_i}}{\left.\frac{\partial D(z)}{\partial z}\right|_{z=p_i}}$$

Si calculamos el numerador y el denominador de esta última expresión tenemos, que de la definición de D(z) obtenemos,

$$\left.\frac{\partial D(z)}{\partial a_k}\right|_{z=p_i} = z^{-k}\Big|_{z=p_i} = p_i^{-k}$$

y

$$\left.\frac{\partial D(z)}{\partial z}\right|_{z=p_i} = \frac{\partial}{\partial z}\left(\prod_{k=1}^{N}\left(1-p_k z^{-1}\right)\right)\Bigg|_{z=p_i} = \frac{\partial}{\partial z}\left(\prod_{k=1}^{N}\left(\frac{z-p_k}{z}\right)\right)\Bigg|_{z=p_i} = \sum_{k=1}^{N}\frac{p_k}{z^2}\prod_{\substack{j=1\\ j\neq k}}^{N}\left(1-p_j z^{-1}\right)\Bigg|_{z=p_i} =$$

$$= \sum_{k=1}^{N}\frac{p_k}{p_i^{\,2}}\prod_{\substack{j=1\\ j\neq k}}^{N}\left(\frac{p_i-p_j}{p_i}\right)$$

Solo cuando $k=i$ ninguno de los términos entre paréntesis es cero, por tanto el productorio no se anula. Luego el sumatorio sólo tiene un término.

$$\left.\frac{\partial D(z)}{\partial z}\right|_{z=p_i} = \frac{p_i}{p_i^{\,2}\,p_i^{\,N-1}}\prod_{\substack{j=1\\ j\neq i}}^{N}\left(p_i-p_j\right) = \frac{1}{p_i^{\,N}}\prod_{\substack{j=1\\ j\neq i}}^{N}\left(p_i-p_j\right)$$

Luego la expresión buscada es:

$$\boxed{\Delta p_i = \sum_{k=1}^{N}\frac{p_i^{\,N-k}}{\prod_{\substack{j=1\\ j\neq i}}^{N}\left(p_i-p_j\right)}\Delta a_k}$$

Esta expresión proporciona una medida de la sensibilidad del polo i-ésimo a cambios en el coeficiente a_k.

A partir de esta expresión obtenemos las siguientes conclusiones:

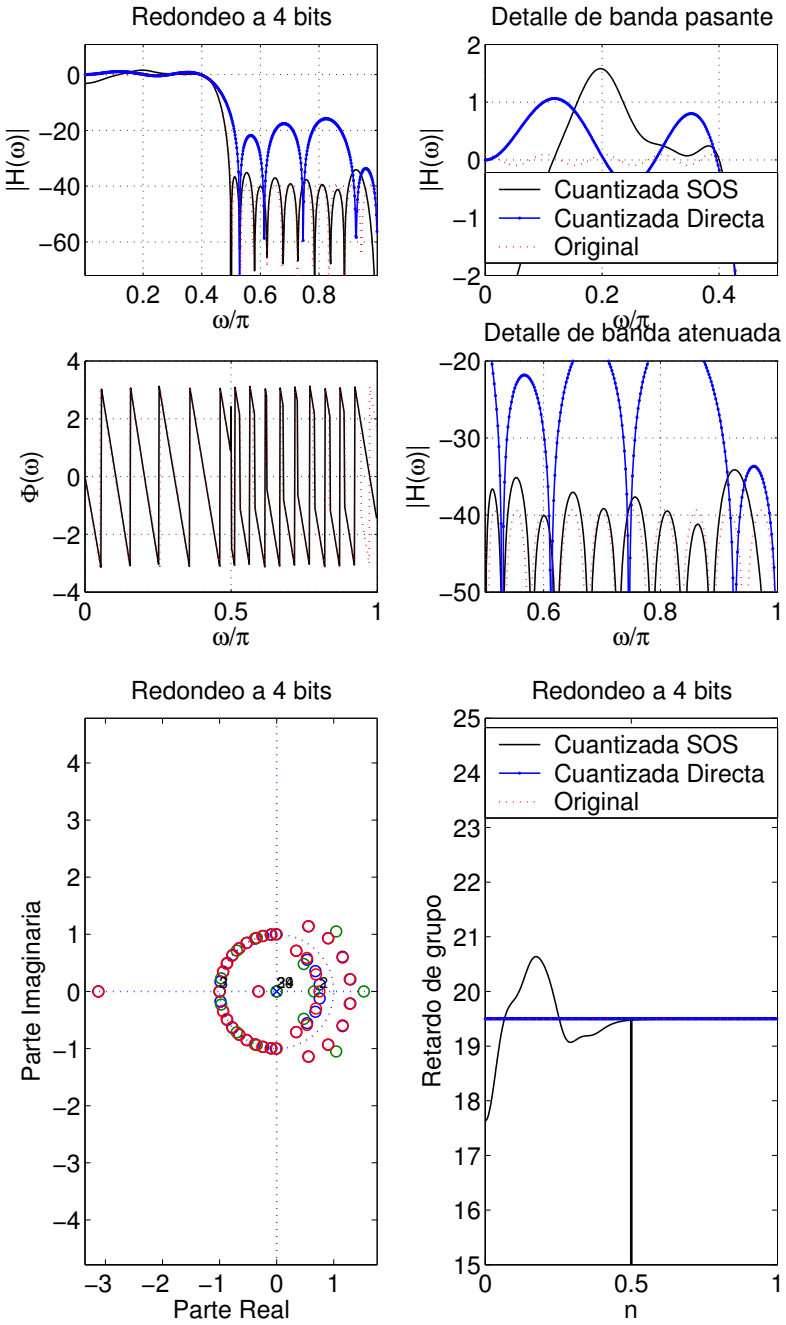
- Como $|p_i|<1,\ |p_i|^N << |p_i|^0 \Rightarrow$ a_N es el coeficiente más sensible; a_0 es el coeficiente menos sensible
- Cuanto mayor sea el orden más sensibilidad.
- La sensibilidad aumenta si los polos están muy cercanos (como el los filtros pasa banda estrechos), ya que el término $\left(p_i-p_j\right)$ es la distancia entre el polo considerado y los restantes. Utilizaremos **descomposiciones en cascada** para disminuir el orden. Idealmente de orden 1, pero para evitar aritmética compleja agruparemos en **etapas de 2º orden,** además los polos complejos conjugados están suficientemente alejados entre sí, lo que disminuye la sensibilidad.

6.6.2.- Sensibilidad frente a la cuantización de los ceros en filtros FIR.

Un estudio análogo al realizado para los polos se puede llevar a cabo para los ceros, que nos permitiría obtener una expresión análoga para la sensibilidad de los ceros en una implementación directa. La principal diferencia es que los filtros FIR, en su mayoría, se diseñan para tener fase lineal, como consecuencia $b_k = \pm b_{M-1-k}$. Luego la cuantización de los coeficientes en forma directa sigue preservando la linealidad, que no ocurre si descomponemos en etapas de 2º orden (sí se cumple para etapas de cuarto orden agrupando un par de ceros complejos conjugados y sus recíprocos). Es decir, la cuantización únicamente afecta a la respuesta en frecuencia en módulo. Por otra parte, al mantener la linealidad, los ceros que originalmente se encuentran sobre la circunferencia unidad, al cuantizar seguirán en la misma posición (de lo contrario se perdería la linealidad de fase). Además los ceros situados sobre la circunferencia unidad se verán afectados por igual al cuantizar los coeficientes b_k, dado que el término[3] $z_i^{M-1-k} = 1$ si $|z_i| = 1$. Por estas razones la estructura directa sí se prefiere para filtros FIR.

En las siguientes figuras mostramos las gráficas obtenidas para una descomposición en cascada con etapas de segundo orden del filtro FIR descrito en el apartado 6.6.

[3] La expresión obtenida es la misma que para los polos $\Delta z_i = \sum_{k=1}^{M-1} \frac{z_i^{M-1-k}}{\prod_{\substack{j=1 \\ j\neq k}}^{M-1} (z_i - z_j)} \Delta b_k$ con $H(z) = \sum_{k=0}^{M-1} b_k z^k$



6.6.3.- Cuantificación de las formas en paralelo y en cascada.

Hemos indicado que un filtro IIR de orden superior a 2 se debe implementar como una combinación de secciones de segundo orden, pero no hemos indicado si se debe emplear

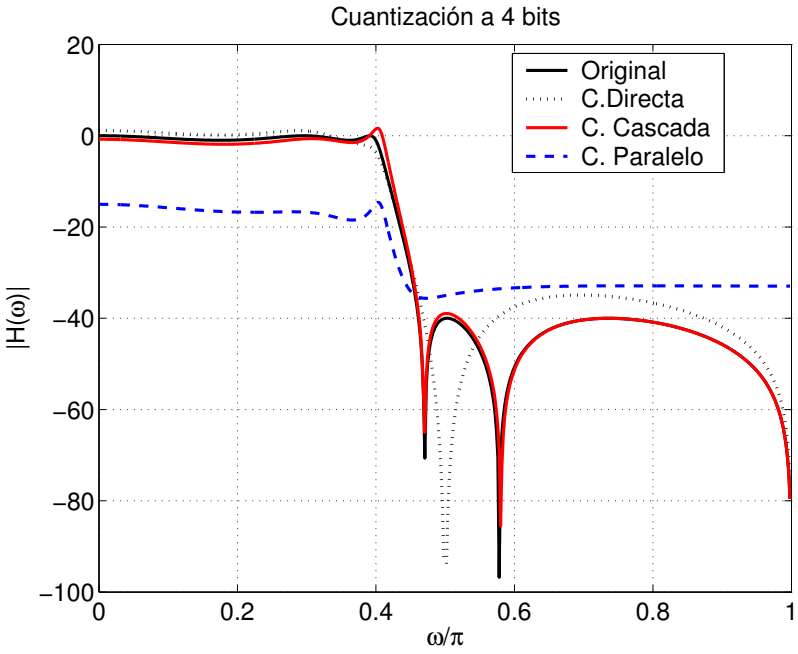
una configuración en cascada o en paralelo. Es decir, hemos de elegir entre $H(z)=\prod_{k=1}^{K}\frac{b_{k0}+b_{k1}z^{-1}+b_{k2}z^{-2}}{1+a_{k1}z^{-1}+a_{k2}z^{-2}}$ o $H(z)=\sum_{k=1}^{K}\frac{c_{k0}+c_{k1}z^{-1}}{1+a_{k1}z^{-1}+a_{k2}z^{-2}}$.

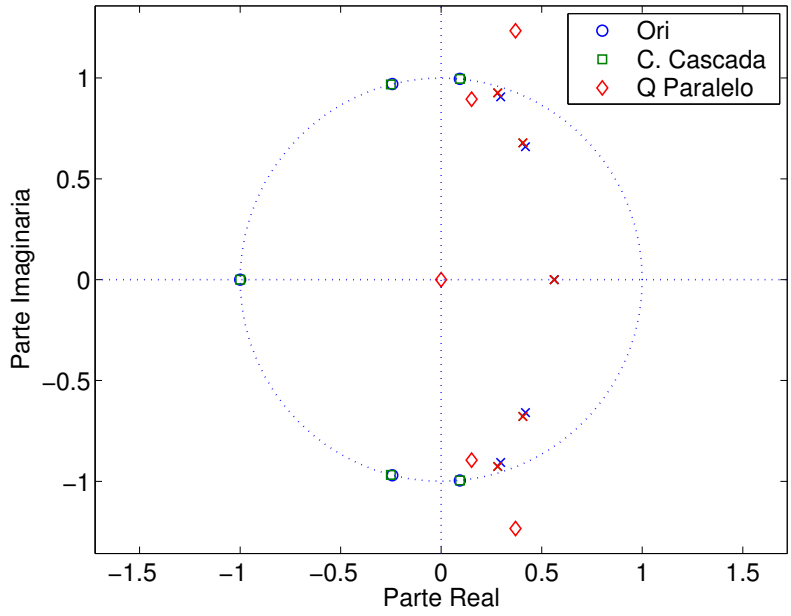
Si el filtro se implementa como una descomposición en cascada de términos de primer y segundo orden, la relación entre los ceros y los coeficientes cuantizados es directa ya que la cuantizacion de los coeficientes sólo afecta a 2 ceros complejos conjugados (los de esa etapa), sin embargo cuando se opta por una descomposición en paralelo, en el cálculo de los coeficientes c_{k0}, c_{k1}, intervienen todos los coeficientes b_k; es decir, la cuantización de c_{k0}, c_{k1} afectará a TODOS LOS CEROS.

Esto hace que sea difícil evaluar la perturbación introducida en la posición de los ceros, haciendo que la sensibilidad aumente. Además, si el filtro tiene ceros sobre la circunferencia unidad, es probable que tras la descomposición en paralelo dejen de estarlo. Esto no ocurre cuando tenemos ceros sobre la circunferencia unidad en una descomposición en cascada ya que el coeficiene b_{2k} será la unidad por lo que no se verán afectados por la cuantización; los ceros podrán experimentar ligeras modificaciones pero siempre se mantendrán sobre la circunferencia unidad.

En el caso que sea necesario una implementación en paralelo se puede optar por una representación en coma flotante.

La forma en cascada es la más robusta frente a la cuantización de los coeficientes y debe ser la elección ante una implementación en coma fija.





6.6.4.- Importancia de la estructura frente a la cuantización.

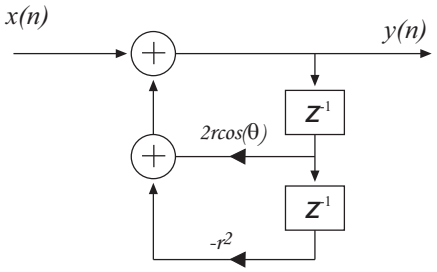
Del apartado anterior hemos concluido que para un filtro IIR la forma más adecuada de implementación es mediante etapas en cascada de 1er y 2º orden, pero no hemos especificado nada a cerca del tipo de estructura empleada para cada etapa.

Si consideramos un sistema de primer orden, $H(z)=\frac{1}{1-az^{-1}}$ y cuantizamos con b bits la posición del polo $a = L\Delta \quad \Delta = 2^{-b}$; es decir, las posiciones del polo están equiespaciadas a intervalos Δ en el eje real.

Para un sistema de 2º orden:

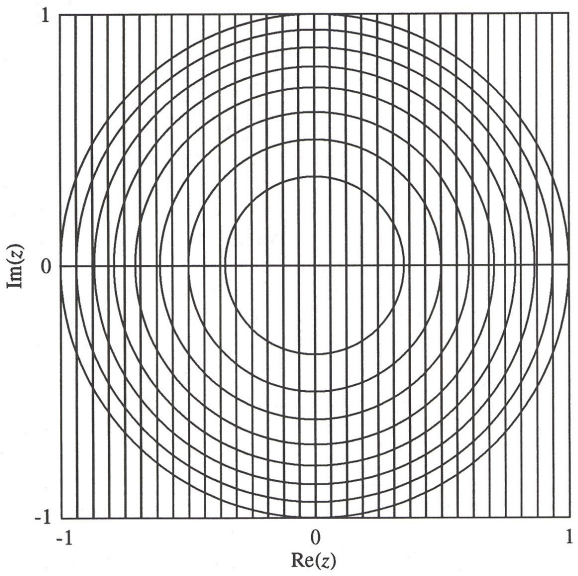
$$H(z)=\frac{1}{1+a_1 z^{-1}+a_2 z^{-2}} \qquad \begin{matrix} a_1 = -2r\cos(\theta) \\ a_2 = r^2 \end{matrix} \qquad \text{polos en } z = r\cdot e^{\pm j\theta}$$

si cuantizamos la forma directa



Al cuantizar los coeficientes a_1 y a_2 estamos cuantizando la parte real de los polos y la distancia de los polos al origen r^2. Si b=4 (3 bits fraccionales más uno de signo) las posibles localizaciones de los polos son las mostradas en la siguiente figura. Observamos que en el eje X la distribución de la cuantización es uniforme ya que el coeficiente coincide con la parte real, pero el eje Y no estamos cuantizando la parte imaginaria sino r^2.

$$\left.\begin{matrix} a_1 = L_1 2^{-b} \\ a_2 = L_2 2^{-b} \rightarrow r = \sqrt{L_2 2^{-b}} \end{matrix}\right\} L_1, L_2 \text{ números enteros}$$



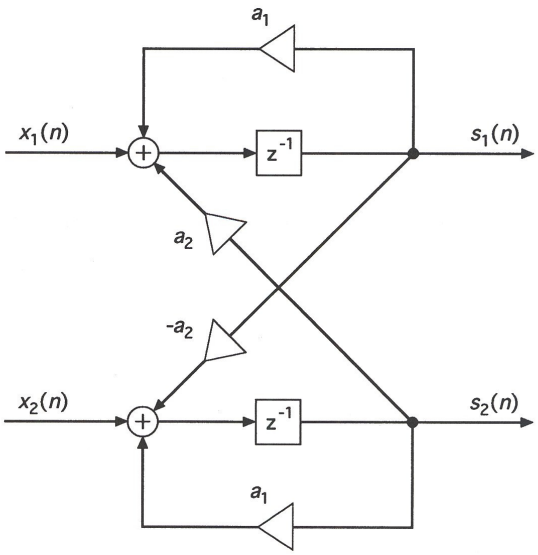
Extraído de: Digital Filters and Signal Processing. L. B.Jackson

Las posiciones válidas para los coeficientes cuantizados son las intersecciones de las líneas verticales y las circunferencias. Observamos que la malla es muy poco densa en $z = \pm 1$ comparado con $z = \pm j$, luego:

Si se diseñan filtros muy estrechos, pasa-baja o pasa-alta, cuyos polos están en $z = \pm 1$, esta estructura será muy sensible a la cuantización por lo que será necesario utilizar mayor precisión (representar con mayor número de bits) o bien BUSCAR OTRA ESTRUCTURA.

Este efecto se produce también cuando se utiliza la técnica de *oversampling*; es decir, muestrear a una frecuencia muy superior a la dictada por el teorema de muestreo, ya que los coeficientes de los filtros toman valores tales que los polos se aproximan a $z = \pm 1$, incrementándose la sensibilidad.

Una alternativa es utilizar otra estructura para la implementación del sistema de segundo orden como es la FORMA ACOPLADA O NORMAL que mostramos en la figura siguiente.



Extraído de: Digital Filters and Signal Processing. L. B.Jackson

Consideremos:

$x_1(n) = x(n)$

$s_2(n) = y(n)$. También podemos considerar como salida $s_1(n)$ (No afecta a los polos.)

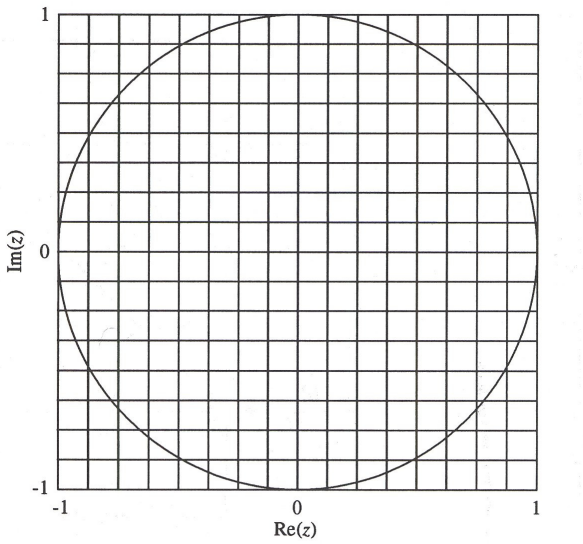
$x_2(n) = 0$

Las ecuaciones del sistema son:

$$\left.\begin{aligned} S_1(z) &= z^{-1}\left(a_1 S_1(z) + a_2 Y(z) + X(z)\right) \\ Y(z) &= z^{-1}\left(-a_2 S_1(z) + a_1 Y(z)\right) \end{aligned}\right\} \begin{aligned} &\rightarrow S_1(z)\left[1 - a_1 z^{-1}\right] = z^{-1}\left[a_2 Y(z) + X(z)\right] \\ &\rightarrow Y(z)\left[1 - a_1 z^{-1}\right] = -a_2 z^{-1} S_1(z) \end{aligned}$$

$$H(z) = \frac{-a_2^2 z^{-2}}{1 - 2a_1 z^{-1} + \left(a_1^2 + a_2^2\right) z^{-2}}$$

Si $a_1 = r\cos(\theta), a_2 = r\sin(\theta)$, los polos de la función de transferencia obtenida coinciden con el sistema de 2º orden de partida salvo que en esta estructura los coeficientes a_1 y a_2 son la parte real e imaginaria de los polos por lo que su cuantización da lugar a una rejilla rectangular como se muestra en la siguiente figura.



Extraído de: Digital Filters and Signal Processing. L. B.Jackson

VENTAJAS → CUANTIFICACION UNIFORME

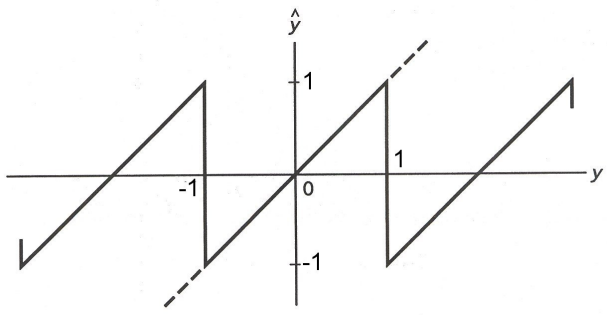
INCONVENIENTE → AUMENTA EL NÚMERO DE PRODUCTOS 2 → 4

Esta estructura es adecuada para el diseño de filtros pasa-baja y pasa-alta estrechos, ya que la densidad de puntos en $z = \pm 1$ es mayor respecto a la estructura directa.

Si por el contrario hemos de implementar filtros pasa-banda o elimina-banda estrechos la estructura directa es más adecuada ya que los polos se encuentran en la zona $z = \pm j$ donde la densidad de puntos es mayor.

6.7.- Escalado de los coeficientes para prevenir desbordamiento.

Cuando se utiliza aritmética binaria fraccional de complemento a 2, la suma de dos números grandes del mismo signo puede originar un valor que exceda del máximo representable, dando lugar a un resultado de distinto signo. En la siguiente figura se muestra la característica funcional de la suma en complemento a 2.



Extraído de: Digital Filters and Signal Processing. L. B.Jackson

En un filtro recursivo, el desbordamiento producido en una suma, puede volver a producirse debido a la realimentación, o bien puede dar lugar a oscilaciones automantenidas difíciles de detener sino se pone a cero el sistema.

En los filtros FIR, dado que no existe realimentación, y que la salida es una suma ponderada de las entradas retardadas, la utilización de aritmética en complemento a 2 tiene la ventaja de que la suma de términos cuyo resultado final no produce desbordamiento, proporcionará un resultado correcto aunque en alguna de las sumas intermedias se produzca.

El desbordamiento se produce a la salida de los sumadores, ya que si las entradas y los coeficientes se representan en formato fraccional binario, éstos se encontrarán en el intervalo $-1 \le x < 1$ por lo que los productos no producirán desbordamiento pero sí pueden producirlo las sumas.

Una forma de prevenir el desbordamiento es ESCALANDO las entradas de los sumadores por un factor adecuado para mantener su salida dentro de los límites permitidos por la representación. Si se realiza un escalado de la señal de entrada por un factor *S (S<1)*, de acuerdo con la expresión $SNR_{AD} = \left(10\log\left(\sigma_x^2\right) + 10.8 + 6.02b\right)dB$, la varianza de la señal escalada será $S^2\sigma_x^2$ y dado que *S<1* producirá un disminución de la SNR. Es decir deberemos elegir un factor de escalado que suponga un compromiso entre la reducción de la SNR y que evite el desbordamiento.

Los métodos más habituales para el cálculo de los coeficientes de escalado son los que se indican a continuación. En todos los casos $f(k)$ es la respuesta impulsional del filtro $F(z)$. Siendo $F(z)$ la función de transferencia desde la entrada hasta la salida del nodo sumador considerado.

$$L_1: \qquad S = \sum_{k=0}^{\infty} |f(k)|$$

$$L_2: \qquad S = \left[\sum_{k=0}^{\infty} f^2(k) \right]^{1/2}$$

$$L_\infty: \qquad S = \max |F(\omega)|$$

$$L_2 < L_\infty < L_1$$

L_1: es la condición más restrictiva y asegura que no se va a producir desbordamiento a la salida, si bien en muchas ocasiones proporciona un escalado excesivo, disminuyendo la SNR.

L_2: en este caso se imponen restricciones sobre la energía de la entrada y la función de transferencia. Para algunas estructuras es posible obtener expresiones compactas ya que

$$s = \sum_{k=0}^{\infty} f^2(k) = \frac{1}{2\pi j} \oint_C F(z) F(z^{-1}) z^{-1} \, dz$$

(Se trata de una integral de contorno $|z| = 1$ que se resuelve mediante el Teorema de los Residuos)

(Ver Mitra-2001)

L_∞: $F(\omega)$ es la respuesta en frecuencia desde la entrada hasta la salida del sumador. Este factor de escalado es el valor de pico de la respuesta en frecuencia. Asegura que no se va a producir desbordamiento cuando la entrada es una sinusoide pura.

Los factores de escalado verifican:

$$L_2 \leq L_\infty \leq L_1$$

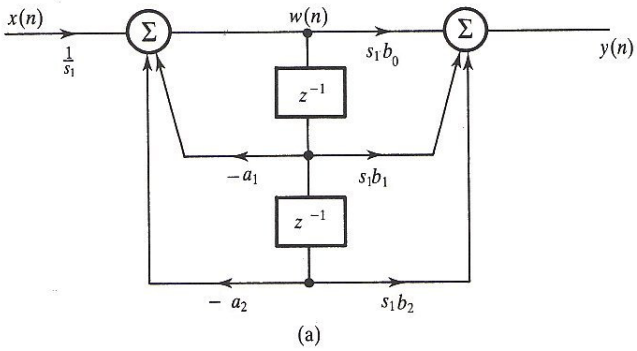
6.7.1.- Escalado de formas de 2º orden.

En la siguiente figura mostramos dos bloques de segundo orden con la forma directa II (a) y forma directa I (b). Al escalar multiplicaremos la entrada por el factor $\frac{1}{s_1}$ y la salida del filtro la multiplicaremos por s_1 para que la función de transferencia total no experimente cambios.

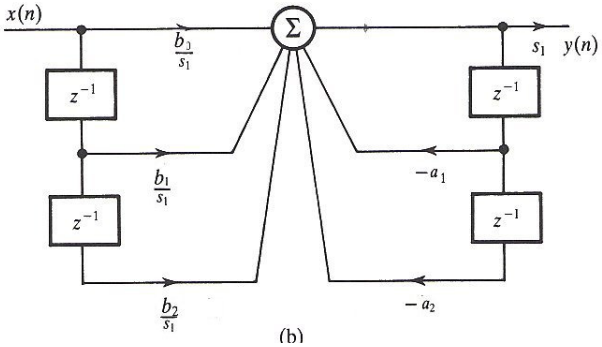
En la forma directa I, tenemos un único sumador que proporcionará la salida del filtro. Como indicábamos anteriormente, si utilizamos aritmética de complemento a 2, si el

resultado de las múltiples sumas no produce desbordamiento, aunque las sumas parciales lo produzcan, el resultado será correcto por lo que no será necesario escalar. Para la forma directa II, la función de transferencia F(z) será:

$$F(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$



(a)



(b)

Extraído de: Digital Signal Processing: a practical aproach. E.C. Ifeachor, B.W.Jervis

6.7.2.- Escalado de formas de formas en cascada y paralelo.

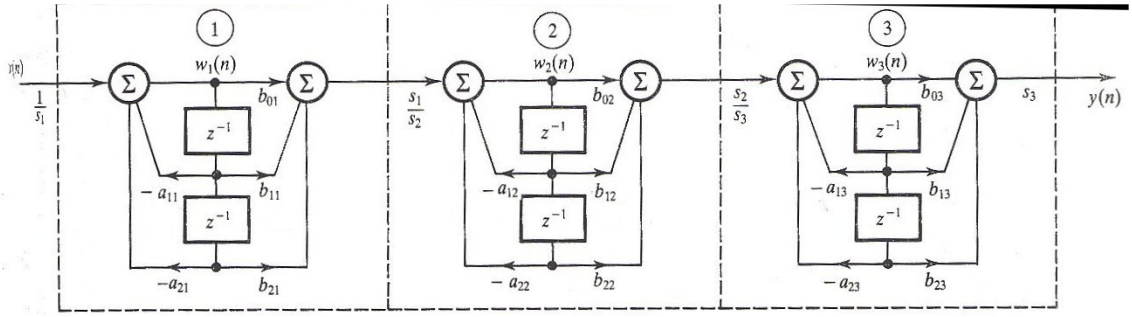
Conexión en cascada:

El esquema es el mismo que en el caso anterior, si bien tendremos un factor de escalado para cada etapa. Si aseguramos que el nodo $w_3(n)$ no produce desbordamiento, el resultado final no lo producirá.

En esta estructura, para el cálculo de los factores de escalado intervienen todas las etapas anteriores; es decir, la funcion de transferencia de la etapa i-ésima $F_i(z)$ es:

$$s = \|F_i(z)\|_p \qquad p = 1,2,\infty$$

$$F_i(z) = \frac{\prod_{k=1}^{i-1} H_k(z)}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \qquad N: \text{número de etapas} \qquad H_k(z) = \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$



Extraído de: Digital Signal Processing: a practical aproach. E.C. Ifeachor, B.W.Jervis

Los factores de escalado de las etapas 2ª y 3ª se suelen incluir en los coeficientes b_k de la etapa anterior para disminuir el número de productos.

Conexión en paralelo:

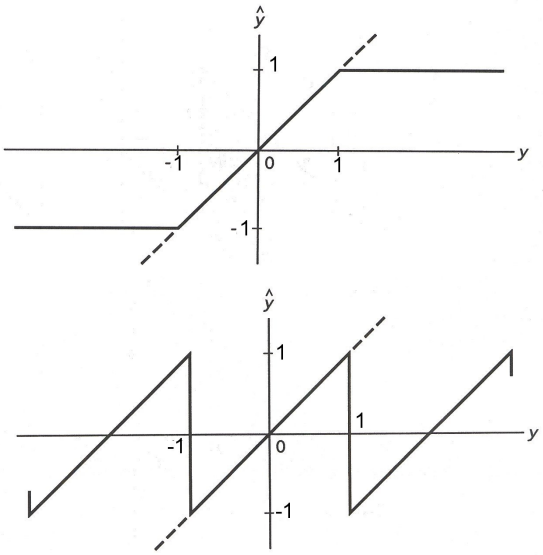
El procedimiento es idéntico al anterior, si bien no hay influencia entre etapas. Para el cálculo del factor de escalado para cada etapa las funciones de transferencia serán[4]:

$$F_i(z) = \frac{1}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \qquad 1 \le i \le N^{\circ}\,\text{ramas}$$

6.7.3.- Detección y prevención del desbordamiento a la salida del filtro.

Cuando utilizamos los factores de escalado determinados por las normas L_2 y L_∞ no se evita completamente la posibilidad de desbordamiento (esto sólo ocurre con L_1). Si utilizamos aritmética en complemento a 2, el desbordamiento hace que la salida cambie bruscamente entre los niveles máximo y mínimo. Para evitar este efecto, se utiliza aritmética saturada. En este caso cuando se produce desboradamiento en las operaciones, la señal de salida se mantiene en el valor máximo o mínimo (dependiendo del tipo de desbordamiento). La siguiente gráfica muestra el resultado de las operaciones con aritmética saturada (superior) y no saturada.

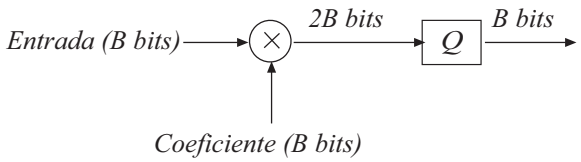
[4] Si alguna de las etapas fuese de primer orden el término $a_{2i} = 0$



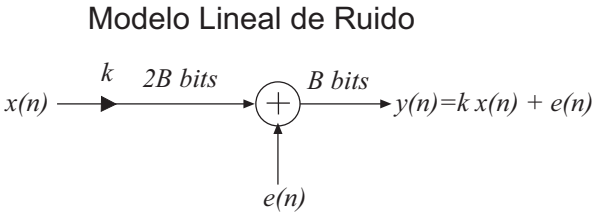
Extraído de: Digital Signal Processing: a practical aproach. E.C. Ifeachor, B.W.Jervis

6.8.- Errores de redondeo en las operaciones producto.

Además de los problemas de desbordamiento que se pueden producir en los nodos suma y como mediante el escalado pueden disminuirse sus efectos, otro factor que hemos de tener en cuenta en la implementación de filtros en coma fija es la cuantización de las operaciones producto. El esquema es el siguiente:



Hemos de cuantizar la salida de 2B bits para obtener un resultado de B bits. Este proceso introduce un error modelizado linealmente como:



El sistema se comporta como un multiplicador ideal al que se le suma una perturbación aleatoria *e(n)* que simula el efecto de la cuantización.

A la secuencia *e(n)*, se le suponen las mismas características que citamos en la conversión AD.

- *e(n)* es una versión muestreada de un proceso estacionario de ruido blanco uniformemente distribuida en el rango de variación del error.
- *e(n)* no está correlacionada con la entrada *x(n)* ni con cualquier otra fuente de error en otro multiplicador[5].

Obtuvimos que la energía del ruido de cuantización venía dada por su varianza cuyo valor es:

$$\sigma_q^2 = \frac{\Delta^2}{12} \qquad \Delta = 2^{-b} \qquad b : \text{bits de la representación (sin signo)}$$

El ruido de redondeo que se produce en cada multiplicador, en la conversión AD, se va a propagar a través del filtro produciendo una señal de ruido a la salida que se solapará con la salida ideal del filtro.

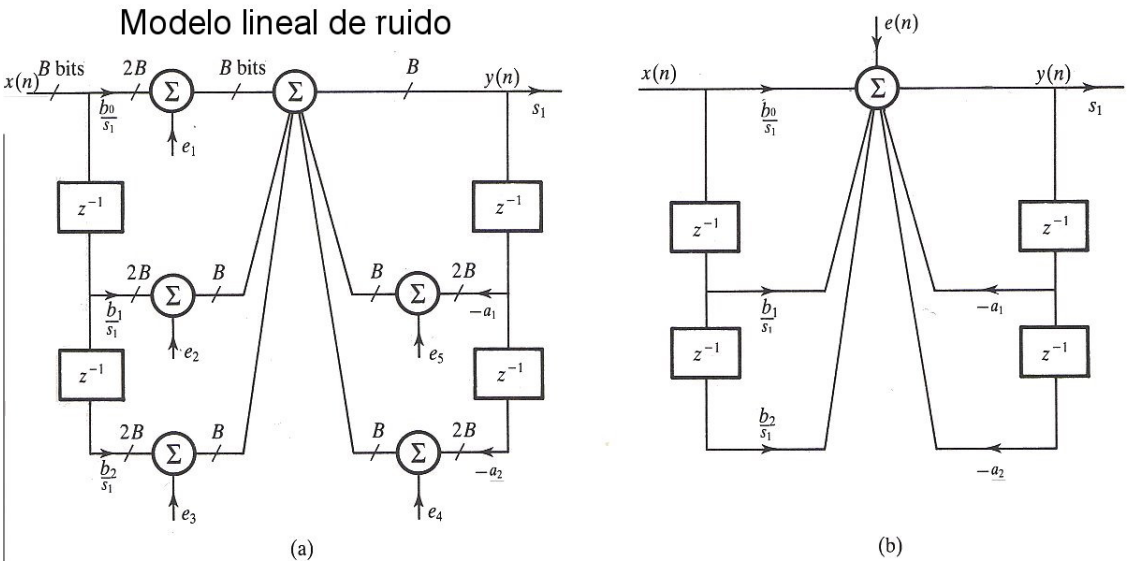
En el análisis siguiente nos aparecerán las expresiones $G_i(z)$ y $g_i(n)$ cuyo significado es el siguiente:

$G_i(z)$: función de transferencia desde la señal de error hasta la salida del filtro.

$g_i(n)$: respuesta impulsional de $G_i(z)$. ($g_i(n) = Z^{-1}\{G_i(z)\}$)

Analicemos el ruido de redondeo en una estructura de 2º orden previamente escalada e implementada mediante la forma directa I. Vamos a considerar que se produce redondeo en cada operación producto.

[5] Recordar que la no correlación con la entrada solo se verifica para cuntización por redondeo y truncamiento con representación en complemento a 2.



Extraído de: Digital Signal Processing: a practical aproach. E.C. Ifeachor, B.W.Jervis

En la gráfica (b) todas las fuentes de ruido que van a un mismo sumador se han agrupado generando un nivel de ruido que será la suma de cada uno de ellos ya que como hemos dichos son independientes entre sí.

En general, sabemos que el ruido a la salida σ_y^2, está relacionado con el ruido en la entrada σ_x^2 por $\boxed{\sigma_y^2 = \sigma_x^2 \sum_{k=0}^{\infty} h^2(k)}$, en nuestro caso el ruido a la entrada es σ_e^2 y el ruido a la salida, debido a la cuantización de las operaciones producto, en esta estructura vendrá dado por[6]:

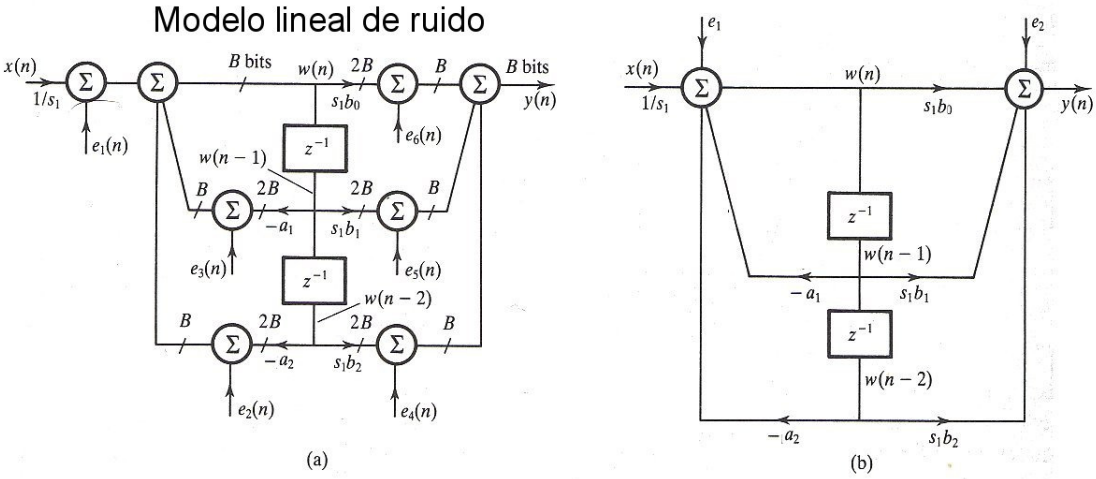
$$\sigma_q^2 = 5\sigma_e^2 \left[\sum_{k=0}^{\infty} g^2(k) \right]$$

$$G(z) = \frac{s_1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

El ruido total presente a la salida del filtro debido a la conversión AD y a la cuantización de las operaciones producto será la suma de ambos. Además si tenemos en cuenta que $\sigma_e^2 = \sigma_{AD}^2 = \frac{\Delta^2}{12}$ obtenemos:

$$\sigma_o^2 = \sigma_{oAD}^2 + \sigma_q^2 = \frac{\Delta^2}{12}\left[\sum_{k=0}^{\infty} h^2(k) + 5\sum_{k=0}^{\infty} g^2(k)\right]$$

El procedimiento es completamente análogo para cualquier estructura. Consideremos una etapa de segundo orden implementada mediante la forma directa II.



Extraído de: Digital Signal Processing: a practical aproach. E.C. Ifeachor, B.W.Jervis

En este caso podemos agrupar las fuentes de ruido en 2 sumadores. e_1 contiene la contribución de 3 productos y e_2 también 3 productos. Difieren en la función de transferencia. La expresión del ruido de cuantización será:

$$\sigma_q^2 = 3\sigma_e^2 \sum_{k=0}^{\infty} g_1{}^2(k) + 3\sigma_e^2 \sum_{k=0}^{\infty} g_2{}^2(k)$$

$$G_1(z) = s_1 \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = s_1 H(z)$$

$$G_2(z) = 1$$

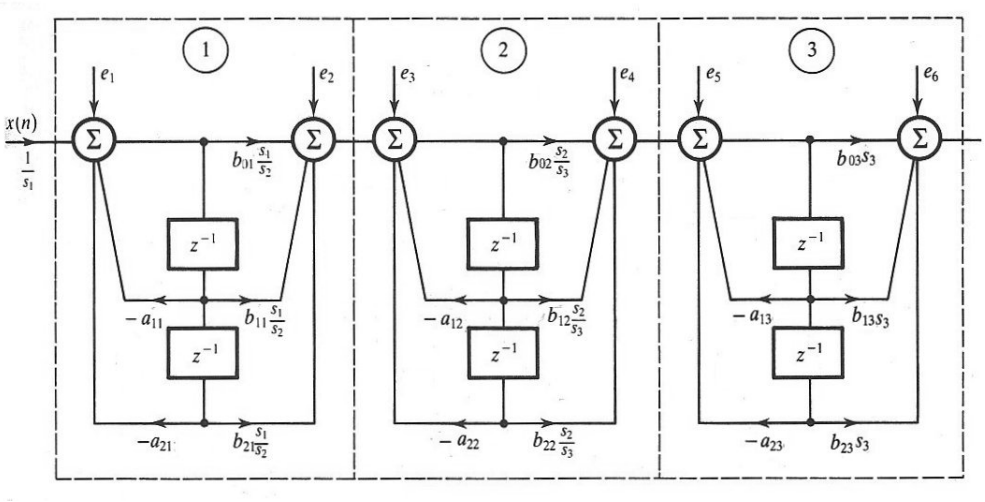
Luego el ruido total a la salida vendrá dado por:

$$\sigma_o^2 = \sigma_{oAD}^2 + \sigma_q^2 = \frac{\Delta^2}{12}\left\{\sum_{k=0}^{\infty} h^2(k) + 3\left[s_1^2 \cdot \sum_{k=0}^{\infty} h^2(k) + 1\right]\right\}$$

[6] Para el cálculo de la función de transferencia $G_i(z)$, solo interviene la entrada $e_i(n)$ el resto de entradas se consideran nulas

Podemos obtener expresiones similares para etapas en cascada y en paralelo.

Conexión en cascada:



Extraído de: Digital Signal Processing: a practical aproach. E.C. Ifeachor, B.W.Jervis

Agrupando los errores en los sumadores obtenemos la expresión para el ruido de cuantización:

$$\sigma_q^2 = \sigma_e^2\left[3\sum_{k=0}^{\infty} g_1{}^2(k) + 5\sum_{k=0}^{\infty} g_2{}^2(k) + 5\sum_{k=0}^{\infty} g_3{}^2(k) + 3\right]$$

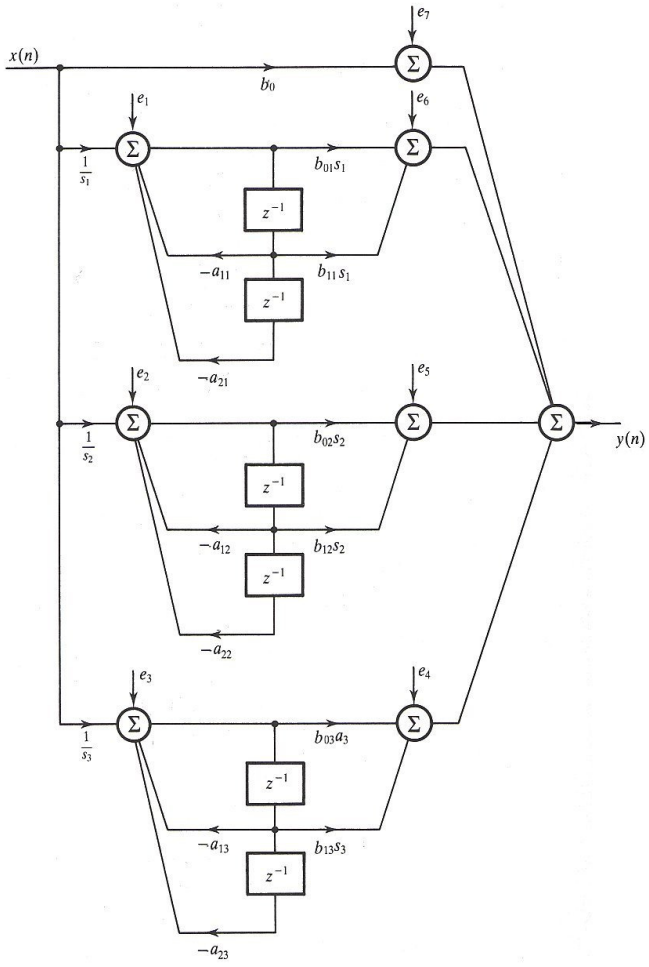
$$G_1(z) = \frac{Y(z)}{E_1(z)} = s_1 \cdot H_1(z) \cdot H_2(z) \cdot H_3(z)$$

$$G_2(z) = \frac{Y(z)}{E_2(z)} = s_2 \cdot H_2(z) \cdot H_3(z)$$

$$G_3(z) = \frac{Y(z)}{E_4(z)} = s_3 \cdot H_3(z)$$

$$G_4(z) = 1$$

Análogamente para la descomposición en paralelo tenemos:



Extraído de: Digital Signal Processing: a practical aproach. E.C. Ifeachor, B.W.Jervis

$$\sigma_q^2 = \sigma_e^2\left[2L+1+3\sum_{i=1}^{L}\sum_{k=0}^{\infty} g_i^{\,2}(k)\right]$$

$$G_i(z) = s_i \cdot H_i(z)$$

El término *2L+1* hace referencia a los 2 productos que cada etapa tiene a la salida, siendo L el número de etapas, más el termino de ganancia. El otro sumando está ligado con los 3 productos que tenemos a la entrada de cada etapa. $H_i(z)$ es la función de transferencia de cada etapa en paralelo.

El estudio realizado considera que se cuantifica la salida de cada operación producto. Los DSP actuales no necesitan redondear cada uno de los productos previo a las sumas ya que disponen de un acumulador de *2B* bits, esto reduce significativamente el nivel de ruido ya

que únicamente se cuantizarán las salidas de los sumadores; es decir, el factor relacionado con el número de productos será siempre la unidad.

Por otra parte, en estas expresiones se observa que los factores de escalado van a producir un incremento muy significativo del nivel de ruido a la salida.

Ejemplo:

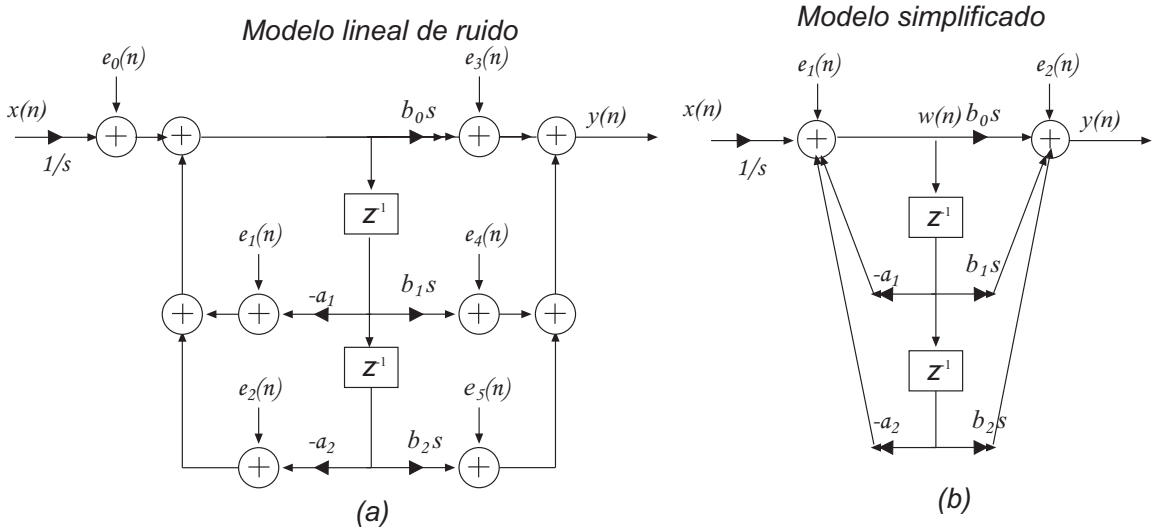
Dado el sistema de segundo orden definido por la función de transferencia

$$H(z) = \frac{0.1436 + 0.2872z^{-1} + 0.1436z^{-2}}{1 - 1.8353z^{-1} + 0.9747z^{-2}}$$

Sabiendo que la estructura utilizada para su implementación es la Forma Directa II determina:

- Factores de escalado L_1, L_2, L_∞.
- Modelo lineal de ruido.
- Estima el ruido total a la salida debido a la cuantización (Operaciones producto y conversión AD)
- Repite el ejercicio considerando la forma directa I.

Solución:



En el modelo (b) los errores están definidos como:

$$e_1(n) \rightarrow \text{suma de errores} \begin{cases} -a_1 \cdot w(n-1) \\ -a_2 \cdot w(n-2) \\ 1/s \cdot x(n) \end{cases} \qquad e_2(n) \rightarrow \text{suma de errores} \begin{cases} b_0 \cdot s \cdot w(n) \\ b_1 \cdot s \cdot w(n-1) \\ b_2 \cdot s \cdot w(n-2) \end{cases}$$

Para el cálculo del factor de escalado, la función de transferencia desde la entrada a la salida del sumador que puede producir desbordamiento (primer sumador) es:

$$F(z) = \frac{1}{1 - 1.8353z^{-1} + 0.9747z^{-2}}$$

Utilizando Matlab podemos calcular los factores de escalado:

```
B=[0.1436, 0.2872 0.1436];
A=[1 -1.8353 0.9747];
%Respuesta impulsional de F(z)
h=impz(1,A);
L1=(sum(abs(h)));
L2=sqrt(sum(h.^2)); %También se puede utilizar la función norm()
[H,w]=freqz(1,A);
Linf=max(abs(H));
```

Obtenemos:

$L_1 = 136.3803 \qquad L_2 = 12.1226 \qquad L_\infty = 104.5884$

Para el cálculo del ruido a la salida necesitamos las funciones de transferencia desde las señales de error a la salida, que para nuestro sistema son:

$$G_1(z) = \frac{Y(z)}{E_1(z)} = s_1 \frac{0.1436 + 0.2872z^{-1} + 0.1436z^{-2}}{1 - 1.8353z^{-1} + 0.9747z^{-2}}$$

$$G_2(z) = \frac{Y(z)}{E_2(z)} = 1$$

El error total a la salida debido a la cuantización vendrá dado por:

$$\sigma_q^2 = 3\sigma_e^2 \left[\sum_{k=0}^{\infty} g_1{}^2(k) + 1 \right]$$

y el ruido debido a la conversión AD a la salida vendrá dado por: $\sigma_{oAD}^2 = \sigma_{AD}^2 \sum_{k=0}^{\infty} h^2(k)$.

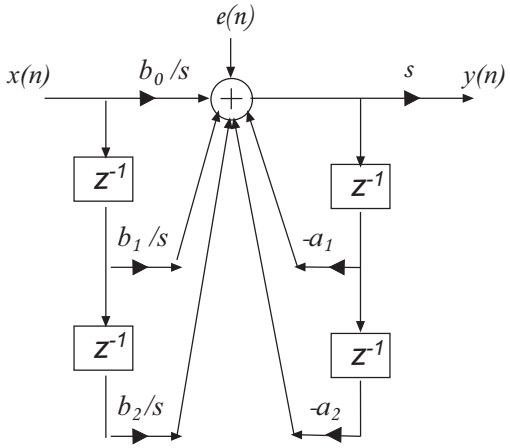
Utilizando Matlab obtenemos,

```
%Ruido a la salida debido a la cuantizacion de las operaciones
%producto
%Consideramos como factor de escalado L2
g1=impz(L2*B,A);
```

```
ruidoq=3*(g1'*g1+1)
h=impz(B,A);
ruidoAD=h'*h;
```

Obtenemos como resultado $\sigma_y^2 = \sigma_q^2 + \sigma_{oAD}^2 = \frac{\Delta^2}{12}[19906.27 + 45.15] = \frac{\Delta^2}{12}19951$

Si utilizamos la forma directa I el modelo de ruido incluyendo el factor de escalado con los coeficientes b_k será:



$$F(z) = \frac{0.1436 + 0.2872z^{-1} + 0.1436z^{-2}}{1 - 1.8353z^{-1} + 0.9747z^{-2}}$$

$L_1 = 75.71 \qquad L_2 = 6.72 \qquad L_\infty = 57.93$

$G(z) = s_1 \frac{1}{1 - 1.8353z^{-1} + 0.9747z^{-2}}$. Si consideramos como factor de escalado L_2

$$\sigma_y^2 = \sigma_q^2 + \sigma_{oAD}^2 = \frac{\Delta^2}{12}[33172.12 + 45.15] = \frac{\Delta^2}{12}33.317$$

En ruido es mayor, pero si tenemos en cuenta que para la forma directa I no es necesario incluir factor de escalado, si se trabaja con aritmética de complemento a 2, obtenemos

$$\sigma_{y\,\text{sin escalar}}^2 = \sigma_q^2 + \sigma_{oAD}^2 = \frac{\Delta^2}{12}[\ 734.79 + 45.15] = \frac{\Delta^2}{12}779.94$$

que es menor que con la estructura directa II.

Ejercicio propuesto: determina el nivel de ruido a la salida del filtro anterior debido a la cuantización si se utiliza la forma directa II traspuesta.

6.9.- Emparejamiento de ceros y polos y ordenación de secciones.

Hemos visto que la forma más adecuada de implementación es la cascada de secciones de segundo orden, y también hemos analizado como determinar los errores producidos a la salida del filtro debidos a la cuantización de las operaciones producto y la necesidad de escalar los coeficientes para evitar el desbordamiento, si bien quedan por resolver dos cuestiones:

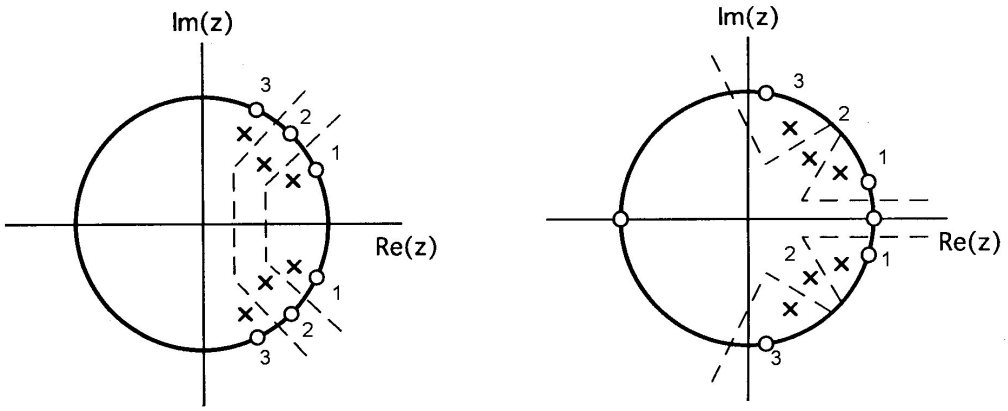
- ¿Cómo agrupamos los ceros y polos para formar las secciones de segundo orden ?
- ¿Cómo ordenamos las secciones de segundo orden en cascada ?

Cada una de las posibles ordenaciones va a tener un nivel de ruido a la salida distinto. Como regla práctica para la ordenación utilizamos la dada por Jackson.

REGLA PRÁCTICA

- En primer lugar, agrupar la pareja de polos complejos conjugados más próxima a la circunferencia unidad con la pareja de ceros complejos más cercana.
- A continuación, agrupar la pareja de polos complejos más próximos a los anteriores con la pareja de ceros complejos más próxima
- Repetir el proceso hasta que todos los ceros y polos estén emparejados

La siguiente figura muestra el orden en que se ha realizado el emparejamiento a partir del diagrama de ceros y polos, de dos sistemas.



Extraído de: Digital Filters and Signal Processing. L. B.Jackson

Este agrupamiento de ceros y polos disminuye el pico de la respuesta en frecuencia de cada una de las secciones de segundo orden, además esto disminuye la posibilidad de desbordamiento y el nivel de ruido a la salida.

Ordenación de secciones.

Una vez obtenidas las secciones de 2º orden hemos de proceder a su ordenación. Si recordamos la definición de las funciones *F(z)* y *G(z)* para el cálculo de los factores de escalado y para determinar el nivel de ruido a la salida respectivamente, las secciones colocadas al principio tendrán mayor influencia en los factores de escalado y la situadas al final influirán más en el cálculo del nivel de ruido a la salida.

Vamos a considerar dos criterios: Minimizar la energía del error a la salida y minimizar el valor de pico del error a la salida (minimiza la posibilidad de desbordamiento). La ordenación depende del tipo de escalado utilizado. La siguiente tabla muestra la ordenación óptima.

Criterio	L_2	L_∞
Minimizar energía error a la salida	Polos cerca de $\|z\|=1$ a polos lejos de $\|z\|=1$ (más picada a menos picada)-*down*-	Orden apenas afecta
Minimizar la posibilidad de **desbordamiento**	Orden apenas afecta	Polos lejos de $\|z\|=1$ a polos cerca de $\|z\|=1$ (menos picada a más picada) –*up*

La función de Matlab *tf2sos* realiza la descomposición en cascada y también la ordenación:

```
[sos,g] = tf2sos(b,a,'order','scale')
scale: 'none',2,inf
order: 'down','up'
```

6.10.- Oscilaciones de ciclo límite en sistemas recursivos

Hasta ahora hemos utilizado un tratamiento de los efectos de trabajar con registros de longitud finita utilizando un modelo lineal. Además hemos supuesto que la señal de entrada no está correlacionada con la secuencia de error.

Un filtro digital es un sistema no lineal, debido a la cuantización de las operaciones, lo cual puede hacer que un sistema estable con precisión infinita pase a ser inestable en precisión

finita para una señal de entrada específica. Este tipo de inestabilidad habitualmente se traduce en un comportamiento periódico a la salida del sistema denominado CICLO LÍMITE.

Un sistema que se encuentra en un ciclo límite permanecerá en esta situación hasta que se aplique una señal de suficiente amplitud como para sacar al sistema de dicho estado.

Los ciclos límite **únicamente se producen en sistemas IIR,** como consecuencia de la recursividad, no en sistemas FIR.

Existen dos tipos de ciclo límite:

- Ciclo límite GRANULAR(CLG). Es un ciclo límite de baja amplitud.
  - CLG Inaccesible: solo se da para un determinado conjunto de condiciones iniciales que caracterizan a dicho ciclo límite
  - CLG Accesible: puede darse aunque las condiciones iniciales no estén dentro del conjunto que caracteriza a dicho ciclo límite.
- Ciclo límite de DESBORDAMIENTO. Es un ciclo límite de gran amplitud

Las amplitudes de salida durante un ciclo límite están confinadas en un intervalo de valores denominado BANDA MUERTA (*Dead Band*) del filtro

Ejemplo: Consideremos un sistema de primer orden $y(n) = x(n) + 0.9y(n-1)$. La respuesta impulsional para $x(n) = 0 \qquad y(-1) = 10$

$y(n) = \{\underset{\uparrow}{9}, 8.1, 7.29, 6.5610, 5.9049, 5.3144, ...\}$

Si en este sistema utilizamos cuantización por redondeo al entero más próximo

$y_q(n) = x(n) + Q[0.9y(n-1)]$ obtenemos

$y_q(n) = \{\underset{\uparrow}{9}, 8, 7, 6, 5, 5, 5...\}$ CICLO LÍMITE DE PERÍODO 1, Banda muerta [0,5]

Si consideramos el sistema $y(n) = x(n) - 0.9y(n-1)$ y realizamos el mismo tipo de cuantización obtenemos:

$y_q(n) = \{\underset{\uparrow}{-9}, 8, -7, 6, -5, 5, -5...\}$ CICLO LÍMITE DE PERÍODO 2, Banda muerta $[-5,5]$.

También se pueden obtener ciclos límite con condiciones iniciales nulas y entrada no nula. En la siguiente tabla se muestra la evolución de la salida del filtro $y(n) = x(n) + \alpha y(n-1)$

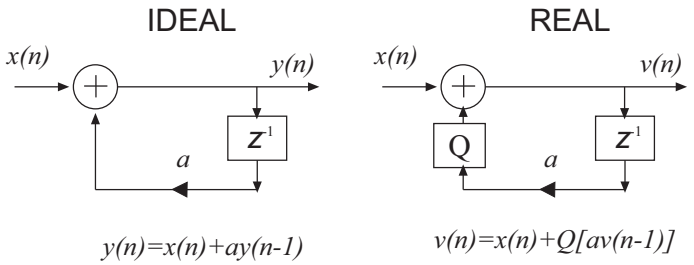
para diversos valores de α ante una entrada $x(n) = 15/16 \cdot \delta(n)$ utilizando cuantización por redondeo y representación signo-magnitud.

n	$a = 0.1000 = \frac{1}{2}$		$a = 1.1000 = -\frac{1}{2}$		$a = 0.1100 = \frac{3}{4}$		$a = 1.1100 = -\frac{3}{4}$	
0	0.1111	$\left(\frac{15}{16}\right)$	0.1111	$\left(\frac{15}{16}\right)$	0.1011	$\left(\frac{11}{16}\right)$	0.1011	$\left(\frac{11}{16}\right)$
1	0.1000	$\left(\frac{8}{16}\right)$	1.1000	$\left(-\frac{8}{16}\right)$	0.1000	$\left(\frac{8}{16}\right)$	1.1000	$\left(-\frac{8}{16}\right)$
2	0.0100	$\left(\frac{4}{16}\right)$	0.0100	$\left(\frac{4}{16}\right)$	0.0110	$\left(\frac{6}{16}\right)$	0.0110	$\left(\frac{6}{16}\right)$
3	0.0010	$\left(\frac{2}{16}\right)$	1.0010	$\left(-\frac{2}{16}\right)$	0.0101	$\left(\frac{5}{16}\right)$	1.0101	$\left(-\frac{5}{16}\right)$
4	0.0001	$\left(\frac{1}{16}\right)$	0.0001	$\left(\frac{1}{16}\right)$	0.0100	$\left(\frac{4}{16}\right)$	0.0100	$\left(\frac{4}{16}\right)$
5	0.0001	$\left(\frac{1}{16}\right)$	1.0001	$\left(-\frac{1}{16}\right)$	0.0011	$\left(\frac{3}{16}\right)$	1.0011	$\left(-\frac{3}{16}\right)$
6	0.0001	$\left(\frac{1}{16}\right)$	0.0001	$\left(\frac{1}{16}\right)$	0.0010	$\left(\frac{2}{16}\right)$	0.0010	$\left(\frac{2}{16}\right)$
7	0.0001	$\left(\frac{1}{16}\right)$	1.0001	$\left(-\frac{1}{16}\right)$	0.0010	$\left(\frac{2}{16}\right)$	1.0010	$\left(-\frac{2}{16}\right)$
8	0.0001	$\left(\frac{1}{16}\right)$	0.0001	$\left(\frac{1}{16}\right)$	0.0010	$\left(\frac{2}{16}\right)$	0.0010	$\left(\frac{2}{16}\right)$

Representación: signo-magnitud. **Cuantización**: redondeo

Extraído de: Tratamiento Digital de Señales. J.G. Proakis

Para un sistema genérico de 1[er] orden tenemos



Cuando la salida del filtro real está en un ciclo límite el sistema se comporta como si tuviese un polo en $z = 1$ para $(a > 0)$ y $z = -1$ para $(a < 0)$ con lo que la salida cuantizada será:

$$Q_r[av(n-1)] = \begin{cases} v(n-1) & a > 1 \\ -v(n-1) & a < 1 \end{cases}$$

Si consideramos cuantización por redondeo el error esta delimitado por:

$$|e_r(n)| \le \frac{\Delta}{2} \qquad |Q_r(av(n-1)) - av(n-1)| < \frac{\Delta}{2}$$

$$|v(n-1)| \le \frac{\frac{\Delta}{2}}{1-|a|} \qquad \Delta = 2^{-b}$$

la expresión de *v(n-1)* define la banda muerta del filtro. Para representación en 4bits+ bits de signo, b=4 y $|a| = 0.5$, la banda muerta es $\left[-\frac{1}{16}, \frac{1}{16}\right]$

Consideremos un sistema de 2º orden.

Sistema lineal ideal: $y(n) = x(n) - a_1 y(n-1) - a_2 y(n-2)$

Sistema no lineal real: $v(n) = x(n) - Q[a_1 v(n-1)] - Q[a_2 v(n-2)]$

Tenemos 2 posibilidades de oscilación:

1. Si consideraremos polos complejos estos se encontrarán en $z = re^{\pm j\theta}$ y $a_1 = -2r\cos(\theta)$ $a_2 = r^2$. Cuando en el ciclo límite, con entrada nula, se verifique $Q[a_2 v(n-2)] = v(n-2)$ el sistema se comportará como si tuviese los polos sobre la circunferencia unidad. En el ciclo límite el error será:

$$|e_r(n)| \le \frac{\Delta}{2} \qquad |Q_r(a_2 v(n-2)) - a_2 v(n-2)| < \frac{\Delta}{2}$$

$$|v(n-2)| \le \frac{\frac{\Delta}{2}}{1-|a_2|} \qquad \Delta = 2^{-b}$$

Expresión que define la banda muerta del filtro (amplitud de la oscilación). La frecuencia de la oscilación viene determinada por $a_1 \cong 2\cos\theta$

2. La otra posibilidad de oscilación con entrada nula, frecuencia 0 ó π, y amplitud v_0 se obtiene cuando al cuantizar $v(n) = x(n) - Q[a_1 v(n-1)] - Q[a_2 v(n-2)]$ se verifica.

$$v_0 = \pm Q[a_1 v_0] - Q[a_2 v_0]$$

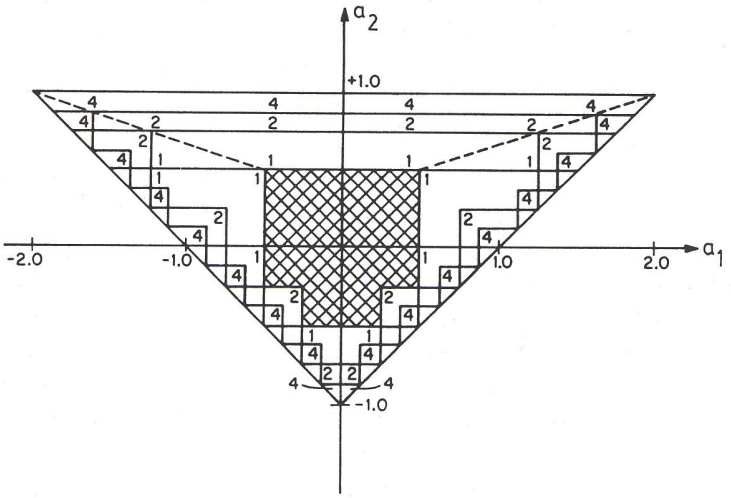
Donde el signo positivo se da para oscilación de frecuencia π.

Si denotamos los errores de cuantización por $e_1(n)$ y $e_2(n)$ podemos obtener que la amplitud de la oscilación al cuantizar por redondeo viene dada por:

$$v_0 = \frac{\Delta}{1-|a_1|+a_2}$$

Si agrupamos las condiciones de estabilidad con las condiciones para la existencia de ciclos límites en sistemas de segundo orden podemos obtener la representación siguiente del

triángulo de estabilidad en el que se han marcado la zona en la que no se producen ciclos límite (rejilla) y las zonas en las que sí se producen así como el período de dicho ciclo límite.



Extraído de: Digital Filters and Signal Processing. L. B.Jackson

Las oscilaciones de ciclo límite de desbordamiento se eliminan en gran medida utilizando aritmética saturada.

Las oscilaciones de ciclo límite granulares se eliminan, si se utiliza una cuantización por truncamiento, en lugar de redondeo, si bien este tipo de cuantización además de presentar *offset* en el valor medio del error, éste está correlacionado con la señal de entrada.

En realizaciones en paralelo, cada sección tienen un comportamiento independiente, pero en etapas en cascada, si la frecuencia del ciclo límite de una etapa se encuentra en las proximidades de la resonancia en etapas posteriores el ciclo límite puede amplificarse, aumentando la banda muerta del filtro.